

Electromagnetic Crack Detection Inverse Problems using Terahertz Interrogating Signals

H. T. Banks ¹, Nathan L. Gibson ¹, William P. Winfree ²

October, 2003

Abstract

We apply an inverse problem formulation to determine characteristics of a defect from a perturbed electromagnetic interrogating signal. A defect (crack) inside of a dielectric material causes a disruption, from reflections and refractions off of the interfaces, of the windowed interrogating signal. We model the electromagnetic waves inside the material with Maxwell's equations. Using simulations as forward solves, our Newton-based, iterative optimization scheme resolves the dimensions and location of the defect. Numerical results are given in tables and plots, standard errors are calculated, and computational issues are addressed.

1 Introduction

The problem we are considering is detecting a crack inside of a dielectric material using electromagnetic interrogation. The idea is to measure the reflected and/or transmitted signals and use the data to solve an inverse problem to determine certain characteristics of the crack, e.g., location and/or width. Possible applications of this procedure include quality assurance in fabrication of certain dielectric materials, or damage detection in aging materials for safety concerns. Further applications of electromagnetic interrogation, as well as additional problem formulations and solutions, can be found in [2].

First we simplify the problem by considering a linearly polarized, pulsed interrogating signal which reduces the problem to one spatial dimension. We start by defining an inverse problem for determining the crack's dimensions. We assume that we have data from sensors, located in front of and/or behind the material, that receive the electromagnetic signal after it is reflected from (or passes through) the material. We then compute simulated signals with approximations to the crack's characteristics and apply an optimization routine to a modified least squares error between this simulated signal and the given data. In our computations we use Maxwell's equations and a Debye polarization equation to model the signal, and solve these equations using a Finite Element method in space and Finite Difference methods in time. Thus the optimization routine finds those crack characteristics which generate a simulated signal that most closely matches the given data. In this sense we determined an estimate to the "true" crack characteristics.

In Section 2 we define the equations that we have chosen in order to model the electromagnetic waves inside the material. We further distinguish between two distinct problem types that we will address, (*Problem 1* and *Problem 2*). Section 3 contains the details of our numerical methods for the simulations. We introduce the inverse problem formulation for *Problem 1* in Section 4, and later improve upon it in Section 4.2. Numerical results of the inverse problem are displayed in Section 4.3.

In Section 5 we begin addressing *Problem 2*. Similarities and differences between the computational issues between the two problems are pointed out. A more sophisticated optimization method is described in Section 5.3 and associated numerical results are given in Section 5.5. Sections 5.5.1 and 5.5.2 explore the effects of adding random noise to the data, both relative and constant variance. In the latter, we compute standard error estimates.

2 Problem Description

We interrogate an (infinitely long) slab of homogeneous nonmagnetic material by a polarized, windowed signal (see [2] for details) in the Thz frequency range (see Figure 1). We start with a wave normally incident

¹Center for Research and Scientific Computation, Raleigh, NC

²NASA Langley, VA.

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE OCT 2003		2. REPORT TYPE		3. DATES COVERED 00-00-2003 to 00-00-2003	
4. TITLE AND SUBTITLE Electromagnetic Crack Detection Inverse Problems Using Terahertz Interrogating Signals				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) North Carolina State University, Center for Research in Scientific Computation, Raleigh, NC, 27695-8205				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES The original document contains color images.					
14. ABSTRACT see report					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES 59	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

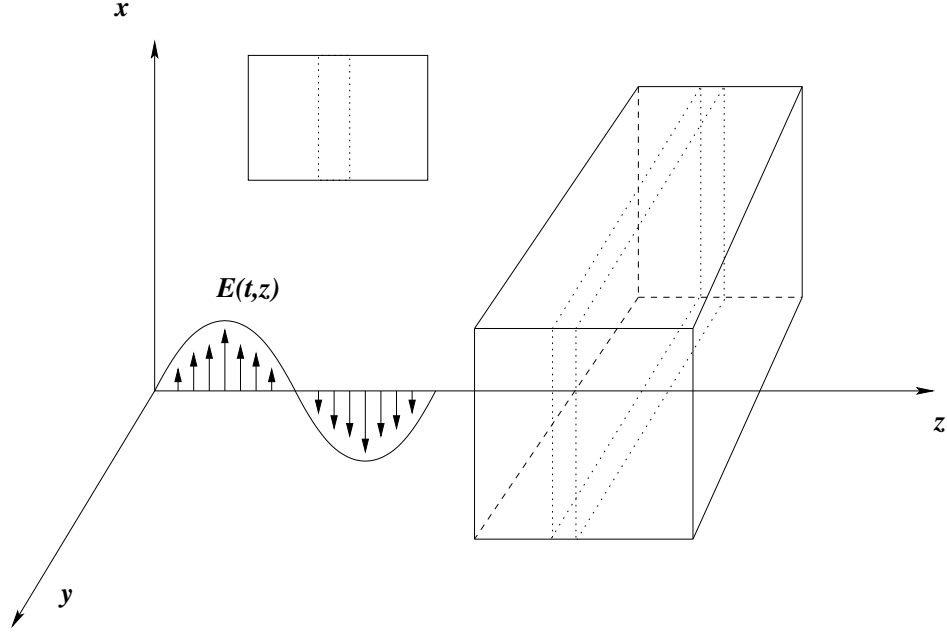


Figure 1: *Problem 1*: Dielectric slab with a material gap in the interior. Possible sensors in front and behind.

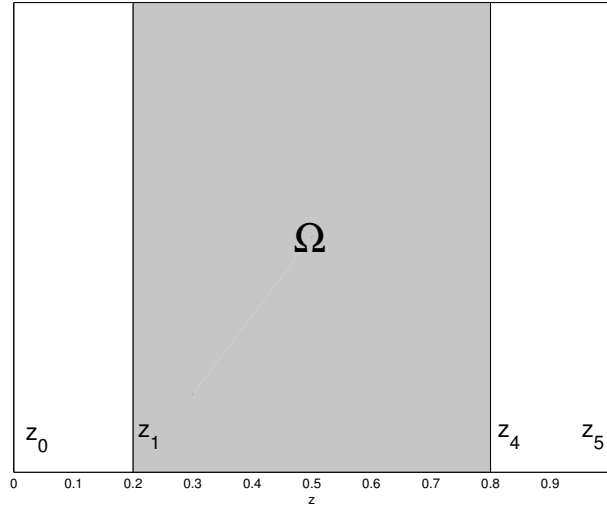


Figure 2: The domain of the material slab: $\Omega = [z_1, z_4]$.

on a slab which is located in $\Omega = [z_1, z_4]$, ($0 < z_1 < z_4 < 1$) with faces parallel to the x - y plane (see Figure 2). We denote the vacuum outside of the material to be Ω_0 . The electric field is polarized to have oscillations in the x - z plane only. Restricting the problem to one dimension, we can write the electric and magnetic fields, \vec{E} and \vec{H} respectively, as follows

$$\begin{aligned}\vec{E}(t, \vec{x}) &= \hat{i}E(t, z) \\ \vec{H}(t, \vec{x}) &= \hat{j}H(t, z),\end{aligned}$$

so that we are only concerned with the scalar values $E(t, z)$ and $H(t, z)$.

Maxwell's equations become:

$$\frac{\partial E}{\partial z} = -\mu_0 \frac{\partial H}{\partial t} \quad (1a)$$

$$-\frac{\partial H}{\partial z} = \frac{\partial D}{\partial t} + \sigma E + J_s, \quad (1b)$$

where $D(t, z)$ is the electric flux density, μ_0 is the magnetic permeability of free space, σ is the conductivity, and J_s is a source current density (determined by our interrogating signal). We take the partial derivative of Equation (1a) with respect to z , and the partial of Equation (1b) with respect to t . Equating the $\frac{\partial^2 H}{\partial z \partial t}$ terms in each, and thus eliminating the magnetic field H , we have:

$$E'' = \mu_0 \left(\ddot{D} + \sigma \dot{E} + \dot{J}_s \right),$$

(where $'$ denotes z derivatives and $\dot{}$ denotes time derivatives).

Note that we have neglected magnetic effects and we have let the total current density be $J = J_c + J_s$, where $J_c = \sigma E$ is the conduction current density given by Ohm's law.

For our source current, J_s , we want to simulate a windowed pulse, i.e., a pulse that is allowed to oscillate for one full period and then is truncated. Further, we want the pulse to originate only at $z = 0$, simulating an infinite antenna at this location. Thus we define

$$J_s(t, z) = \delta(z) \sin(\omega t) I_{[0, t_f]}(t)$$

where ω is the frequency of the pulse, $t_f = 2\pi/\omega$ is fixed, $I_{[0, t_f]}(t)$ represents an indicator function which is 1 when $0 \leq t \leq t_f$ and zero otherwise, and $\delta(z)$ is the Dirac delta distribution.

Remark 1 *Computationally, having a truncated signal introduces discontinuities in the first derivatives which are not only problematic in the numerical simulations (producing spurious oscillations), but are also essentially non-physical. Therefore in our implementation we actually multiply the sine function by an exponential function (see Figure 3) rather than the traditional indicator function. The exponential is of the form*

$$\exp \left(- \left(\frac{t - \theta \frac{t_f}{2}}{\alpha} \right)^\beta \right),$$

where θ controls the horizontal shift, α determines the width and β determines the steepness of the sides. A value of $\theta = 2$ provides a sufficient buffer before the signal to avoid leading oscillations due to the initial discontinuity. For notational consistency we will continue to denote this function as $I_{[0, t_f]}(t)$.

The electric flux density inside the material, given by $D = \epsilon_0 \epsilon_\infty E + P$, is dependent on the polarization, P . Note that ϵ_0 is the permittivity of free space and ϵ_∞ is the relative permittivity in the limit of high frequencies (thus the notation of ∞). For computational testing we assume for this presentation that the media is Debye and thus we use the following polarization model inside Ω :

$$\tau \dot{P} + P = \epsilon_0 (\epsilon_s - \epsilon_\infty) E$$

where ϵ_s is the static relative permittivity and τ is the relaxation time. We also assume $P(0, z) = 0$. Note that in the vacuum outside of Ω , $P = 0$.

In order to represent D in the entire domain, we use the indicator function I_Ω which is 1 inside Ω and zero otherwise. Thus

$$D = \epsilon_0 E + \epsilon_0 (\epsilon_\infty - 1) I_\Omega E + I_\Omega P.$$

In order to have a finite computational domain, we impose absorbing boundary conditions at $z = 0$ and $z = 1$, which are modelled as

$$\begin{aligned} \left[\dot{E} - cE' \right]_{z=0} &= 0 \\ \left[\dot{E} + cE' \right]_{z=1} &= 0. \end{aligned}$$

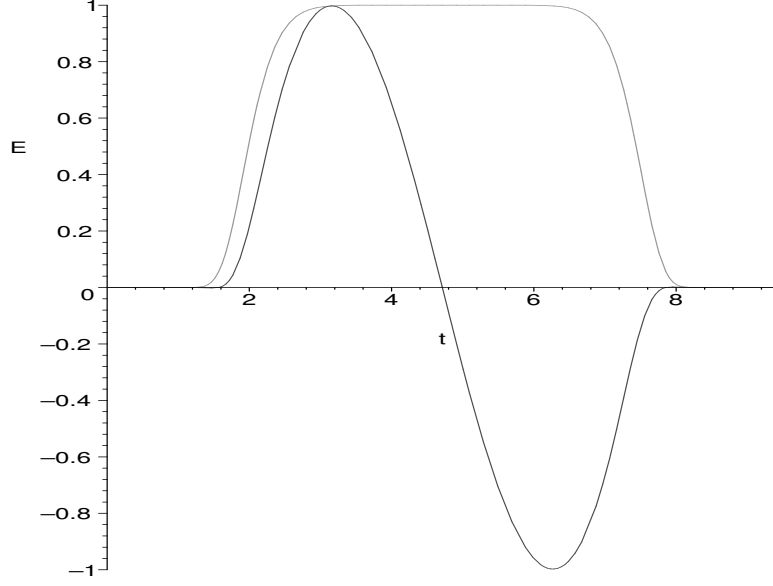


Figure 3: Our choice of a smooth indicator function and the resulting (smoothly truncated) interrogating signal. In this example $\theta = 2$, $\alpha = .9$, and $\beta = 10$.

With these boundary conditions, any incident signal passes out of the computational domain, and does not return. We model this by forcing it to be absorbed by the boundary. Also we assume zero initial conditions, i.e.,

$$\begin{aligned} E(0, z) &= 0 \\ \dot{E}(0, z) &= 0. \end{aligned}$$

Thus our entire system can be written

$$\begin{aligned} \mu_0 \epsilon_0 (1 + (\epsilon_\infty - 1) I_\Omega) \ddot{E} + \mu_0 I_\Omega \ddot{P} + \mu_0 \sigma \dot{E} - E'' &= -\mu_0 \dot{J}_s & \text{in } \Omega \cup \Omega_0 \\ \tau \dot{P} + P &= \epsilon_0 (\epsilon_s - \epsilon_\infty) E & \text{in } \Omega \\ [\dot{E} - cE']_{z=0} &= 0 \\ [\dot{E} + cE']_{z=1} &= 0 \\ E(0, z) &= 0 \\ \dot{E}(0, z) &= 0 \end{aligned}$$

with

$$J_s(t, z) = \delta(z) \sin(\omega t) I_{[0, t_f]}(t).$$

In this formulation we have initially assumed a single slab of a dielectric contained in $\Omega = [z_1, z_4]$. Thus $I_\Omega = 1$ if $z_1 < z < z_4$, and zero otherwise. We now introduce a gap (crack) consisting of a vacuum in the interior of the material as depicted in Figure 4. If the gap is (z_2, z_3) then we redefine $\Omega = \{z | z_1 \leq z \leq z_2 \text{ or } z_3 \leq z \leq z_4\}$. We will refer to this formulation as *Problem 1* (recall Figure 1). Later we will discuss a second formulation, *Problem 2*, where the gap is between the dielectric slab and a metallic (conducting) backing, as shown in Figure 5. This will require slightly different boundary conditions (reflecting instead of absorbing at $z = 1$, where the metal backing begins), but otherwise the numerical methods and analysis are the same.

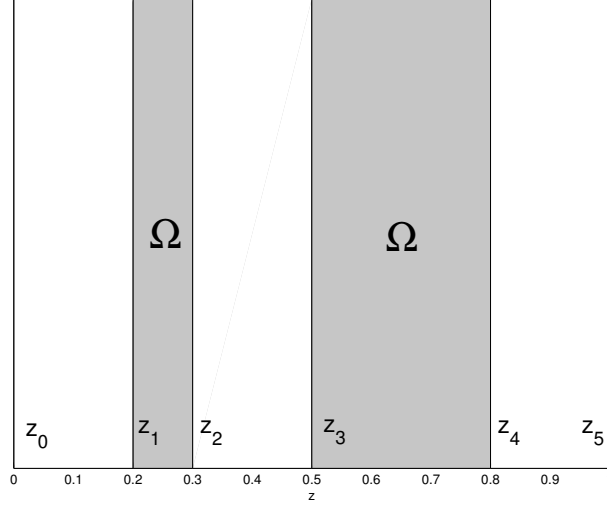


Figure 4: The domain of the material slab with an interior gap between z_2 and z_3 : $\Omega = \{z | z_1 \leq z \leq z_2 \text{ or } z_3 \leq z \leq z_4\}$.

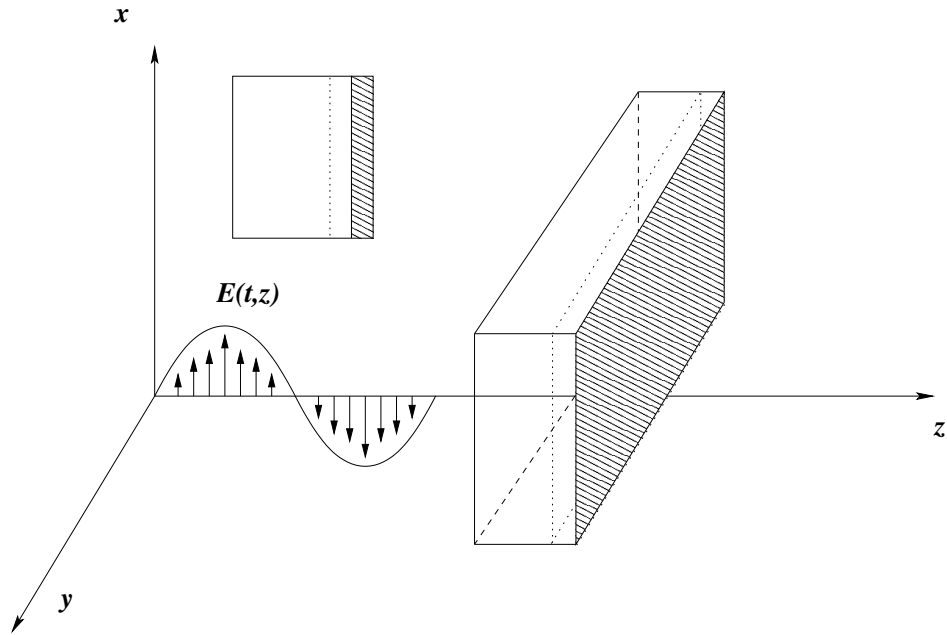


Figure 5: *Problem 2*: Dielectric slab and metallic backing with a gap in between. Possible sensors only in front.

3 Numerical Solution

3.1 Finite Elements

We apply a Finite Element method using standard linear one dimensional basis elements to discretize the model in space. Let N be the number of intervals in the discretization of z , and $h = 1/N$, then the Finite Element discretization has an order of accuracy of $O(h^2)$. For implementation we scale time by $\tilde{t} = ct$ and the polarization by $\tilde{P} = P/\epsilon_0$ for convenience. The resulting system of ordinary differential equations after

the spatial discretization is the semi-discrete form

$$\epsilon_r M \ddot{e} + M^\Omega \ddot{p} + (\eta_0 \sigma M + D + B) \dot{e} + K e = \eta_0 J \quad (2a)$$

$$M^\Omega \dot{p} + \lambda M^\Omega p = \epsilon_d \lambda M^\Omega e, \quad (2b)$$

where $\epsilon_r = (1 + (\epsilon_\infty - 1)I_\Omega)$, $\epsilon_d = \epsilon_s - \epsilon_\infty$, $\lambda = \frac{1}{c\tau}$, and $\eta_0 = \sqrt{\mu_0/\epsilon_0}$. Also e and p are vectors representing the values of E and P respectively at the nodes $\tilde{z}_i = ih$. The mass matrix M has entries

$$M_{ij} = \langle \phi_i, \phi_j \rangle := \int_0^1 \phi_i \phi_j dz$$

where $\{\phi_i\}_{i=1}^N$ are the basis functions (M^Ω is the mass matrix integrated only over Ω), while the stability matrix K has entries

$$K_{ij} = \langle \phi'_i, \phi'_j \rangle := \int_0^1 \phi'_i \phi'_j dz.$$

The matrices D and B result from the boundary conditions where

$$\begin{aligned} D_{1,1} &= 1 \\ B_{N+1,N+1} &= 1 \end{aligned}$$

and all other entries are zero. Finally, J is defined as

$$J_i = - \langle \phi_i, \dot{J}_s \rangle := - \int_0^1 \dot{J}_s \phi_i dz.$$

Note that by differentiating (2b) we can substitute into (2a) and obtain an equation only dependent explicitly on P (two substitutions are required to eliminate \ddot{P} and \dot{P}):

$$\epsilon_r M \ddot{e} + (\eta_0 \sigma M + D + B + \epsilon_d \lambda M) \dot{e} + (K - \epsilon_d \lambda^2 M) e + \lambda^2 M^\Omega p = \eta_0 J.$$

Using shorthand we can write our entire coupled system as

$$M_1 \ddot{e} + M_2 \dot{e} + M_3 e + \lambda^2 \bar{p} = \eta_0 J \quad (3a)$$

$$\dot{\bar{p}} + \lambda \bar{p} = \epsilon_d \lambda M^\Omega e \quad (3b)$$

where $\bar{p} = M^\Omega p$. It is important to mention that each matrix is tridiagonal due to the choice of the linear finite elements.

3.2 Finite Differences

In order to solve the semi-discrete form of our equations we consider two distinct finite difference methods. In the first method we convert the coupled second order system of equations into one larger first order system and simply apply a theta method (unless otherwise stated, we use $\theta = \frac{1}{2}$). In the second method we solve first for the polarization with a forward differencing scheme using the initial conditions and then use that to update a second order central difference scheme for the magnitude of the electromagnetic field. We then continue this process iteratively, alternating between solving for P and for E .

Both methods are second order in time and space for appropriately smooth data (and with $\Delta t = O(h)$). We have compared the errors and the run times of the two methods for several smooth test problems and have determined that the second method is as accurate, but twice as fast, as the first in all cases primarily due to the fact that the linear system is of smaller dimension in the second method. Also, the first method incidentally solves for \dot{e} in addition to e and p , which is superfluous.

In our second method we use a second order central difference scheme to solve (3a). Thus we must first find an approximation to $E(t_1, z)$ where $t_i = i\Delta t$. Note that approximating E with its Taylor expansion around $t_0 = 0$ and applying the initial conditions and ODE, one obtains

$$E(t_1, z) \approx -\frac{\Delta t^2}{2} \mu_0 \dot{J}_s(0, z).$$

Our approach is to first solve for \bar{p} using a θ -method, and then use that approximation to solve for e at the next time step. Thus, our finite difference approximation for (3b) is

$$\bar{p}_{n+1} = \bar{p}_n + \frac{\lambda \Delta t}{1 + \lambda \Delta t \theta} (\epsilon_d M^\Omega e_{n+\theta} - \bar{p}_n) \quad (4)$$

where $[e_n]_j = E(t_n, \tilde{z}_j)$, $[\bar{p}_n]_j = M^\Omega P(t_n, \tilde{z}_j)$, $\tilde{z}_j = jh$, and $e_{n+\theta} = \theta e_n + (1 - \theta)e_{n+1}$ is a weighted average of e_n and e_{n+1} for relaxation to improve the stability of the method. Once we have \bar{p}_{n+1} we can solve for e_{n+2} . Applying second order central difference with averaging to (3a) gives

$$\bar{M}_2 e_{n+2} = \bar{M} e_n + \bar{M}_1 e_{n+1} + \Delta t^2 \eta_0 J_{n+1} - \lambda^2 \Delta t^2 \bar{p}_{n+1}. \quad (5)$$

Note that in this case \bar{M}_2 is tridiagonal and the matrix is the same for each time step, so we may store the Crout LU factorization and use back substitution to solve the system at each time step. For tridiagonal matrices the factorization and the back substitution are both order $O(N)$.

Remark 2 *One point to consider with this method is that if Δt is too large, or too small, we obtain instability in the method which can result in noisy oscillations before, or after, the actual signal. In our testing we have determined that there is an optimal range of values for Δt given a fixed h .*

The inverse problem has difficulties if the sample time of the given data (which is $st := 1/sr$, where sr is the sample rate with units of number of observations per second) is smaller than the time step. In this case we discard the extra data points, effectively multiplying the sample rate by an integer multiple. (We refer to this new sample rate as the effective sample rate sr_e which will be used to determine the observations of the simulations.) Note that since $dt = O(1/N)$, a larger N may result in more data points being retained, and thus provides an additional advantage in finding an optimal solution.

3.3 Method of Mappings

We want the discretization of the domain to incorporate the interfaces between the material and the vacuum so that physical characteristics are not split across more than one finite element. However, we do not want to limit ourselves to simulating cracks that are in effect multiples of a fixed h . Decreasing h to compensate would increase computational time unnecessarily. Instead we employ the “method of mappings”. In essence we map our computational domain to a reference domain with a predetermined discretization, using a linear transformation. In this way we can ensure that all of our boundaries are mapped to node locations. Also, regardless of how small the crack may be, we are guaranteed a fixed number of elements to accurately resolve the behavior of the electric field in that region. Each interval $[\tilde{z}_i, \tilde{z}_{i+1}]$ is mapped to a predetermined interval $[\hat{z}_i, \hat{z}_{i+1}]$ by

$$\hat{z} = \hat{z}_{i+1} \left(\frac{\tilde{z} - \tilde{z}_i}{\tilde{z}_{i+1} - \tilde{z}_i} \right) + \hat{z}_i \left(\frac{\tilde{z} - \tilde{z}_{i+1}}{\tilde{z}_i - \tilde{z}_{i+1}} \right).$$

In effect we are using a variable mesh size, thus we must take care to check that in mapping our domain we do not “stretch” or “shrink” any interval too much because our error is based on the mesh size in the original domain. Although our reference domain may be equally spaced with small subintervals, in general our effective mesh sizes can be quite large. Therefore, we monitor the magnitude of the scaling factors and rediscritize our reference domain if any mesh size is too large. The method of mappings also allows us to easily normalize our domain length, which justifies our theoretical development of the problem in the domain $z \in [0, 1]$.

The method of maps also alters our inner products used in the weak formulation of the problem. We subsequently use a weighted inner product defined as follows:

$$\langle \phi, \psi \rangle = \int_{z_0}^{z_5} \phi(z) \psi(z) dz = \int_0^1 \phi(f^{-1}(\hat{z})) \psi(f^{-1}(\hat{z})) \frac{d\hat{z}}{(f^{-1})'(\hat{z})}$$

where $f(z) = \hat{z}$ is the piece-wise linear transformation on $[z_0, z_5]$.

3.4 Numerical Simulations

The following figures depict the numerical solution of the amplitude of the electric field at various times (Figures 6 and 7), as well as the signal recorded at receivers located at $z = 0$ (Figure 8) and $z = 1$ (Figure 9). We considered a Debye medium with the following parameters:

$$\begin{aligned}\epsilon_s &= 78.2, \\ \epsilon_\infty &= 5.5, \\ \sigma &= 1 \times 10^{-5}, \\ \tau &= 3.16 \times 10^{-8}, \\ f &= 2GHz,\end{aligned}$$

and a material gap located at $[z_3, z_4] = [.6, .61]$.

We have used 2 GHz simply so that our computational domain of $z \in [0, 1]$ would not have to be scaled for this demonstration. Also, in practice, one would not compute a domain so much larger than the material, just as in an experiment the sensors should be as close to the material as possible to reduce noise. We did so here merely so that the full wavelength of the signal would be visible. Finally, we have carried the time length well past the first reflections, again just for demonstration purposes. We chose to plot Figures 8 and 9 with the time unit scaled seconds (ss) which is just seconds multiplied by the speed of light in a vacuum, $ss = c \cdot s$. This, along with the placement of the material $.5m$ from the sensor, validates the timing of the first reflection (see Figure 8) in that it has traveled $1m$ in c seconds.

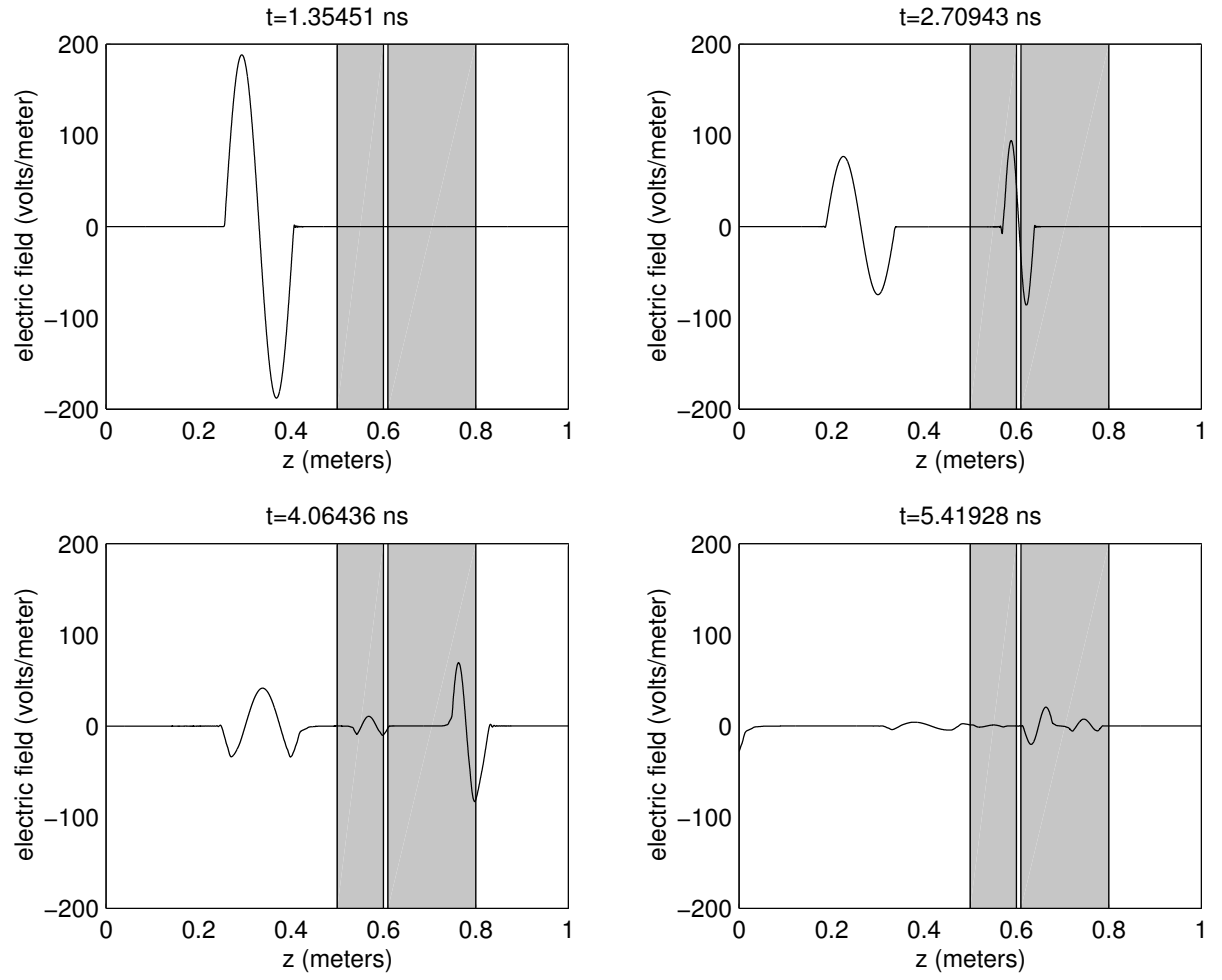


Figure 6: Computed solutions at different times of a windowed electromagnetic pulse incident on a Debye medium with a crack.

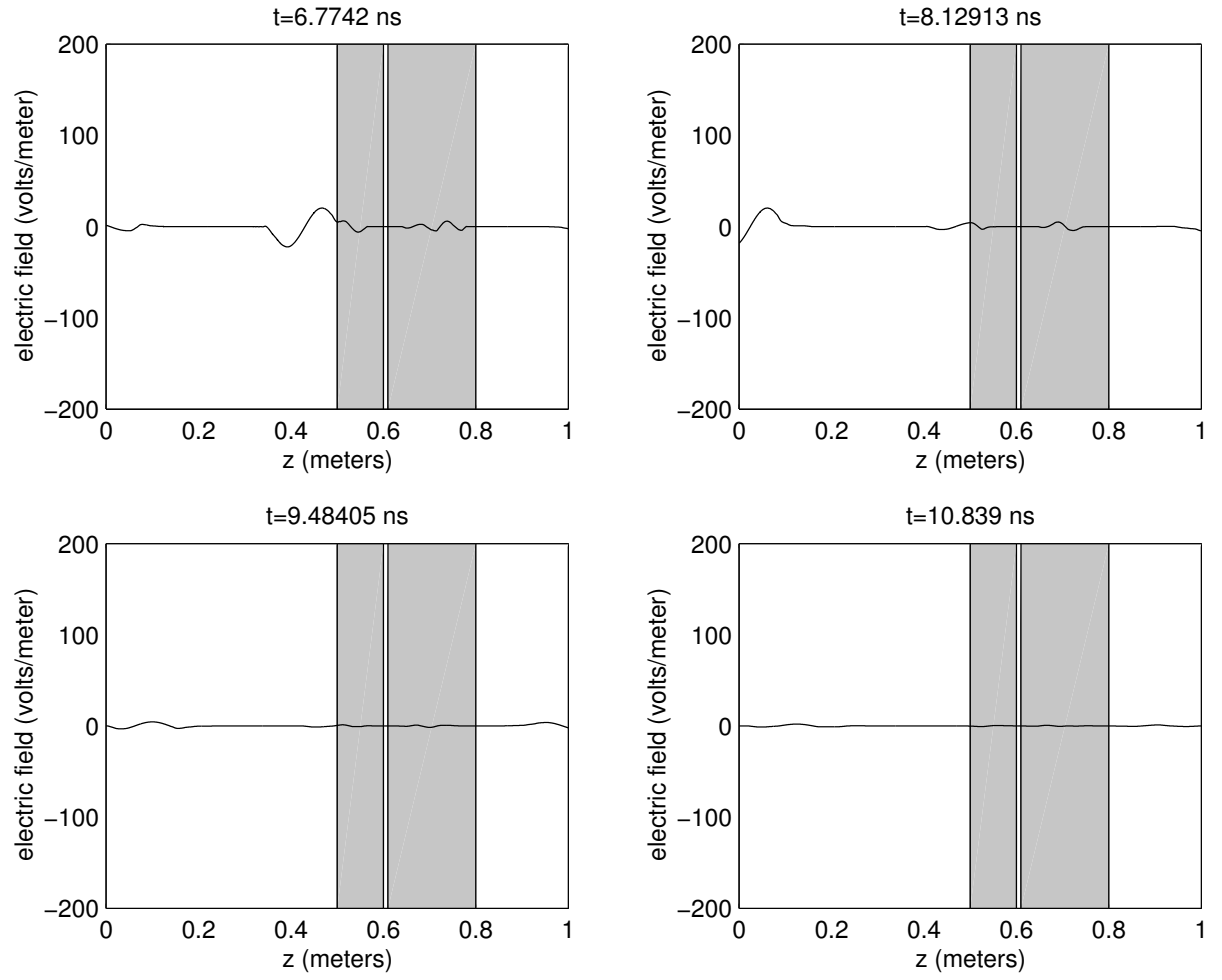


Figure 7: Computed solutions at different times of a windowed electromagnetic pulse incident on a Debye medium with a crack.

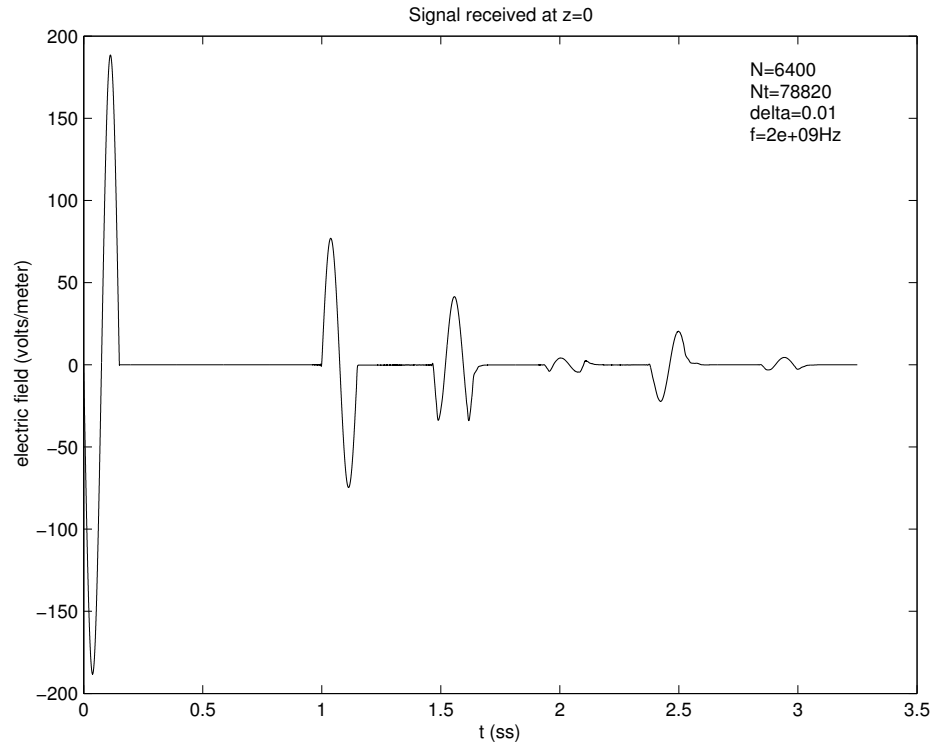


Figure 8: Signal received at $z = 0$, plotted versus seconds scaled by c .

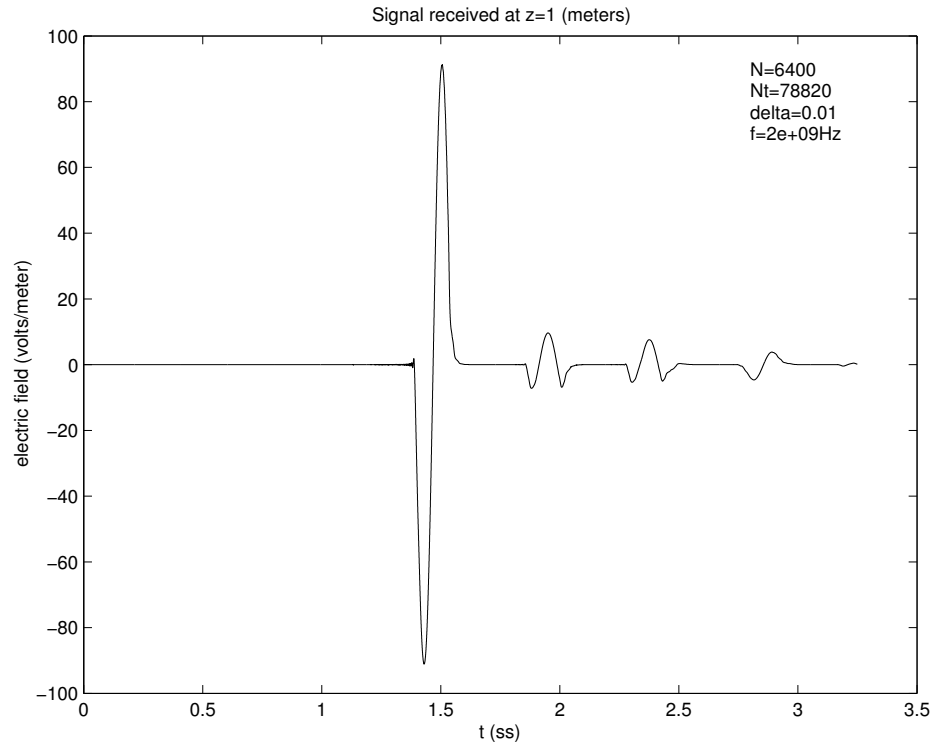


Figure 9: Signal received at $z = 1$, plotted versus seconds scaled by c .

4 Problem 1

We now apply an optimization routine to the least squares error between a simulated signal and the given data to try to determine the crack characteristics. In particular we will be trying to find the depth, $d := z_2 - z_1$, and the width, $\delta := z_3 - z_2$, which will produce a simulated signal most closely similar (in the least squares sense) to the data.

4.1 Inverse Problem

All of the following are solved with respect to a reference problem ($R1$) with these parameter values (see also Figure 4):

$$z_0 = 0, z_1 = .2, z_2 = .3, z_3 = .5, z_4 = .8, z_5 = 1.0$$

$$f = 4 \text{ GHz}, t_f \text{ is one period}$$

$$\tau = 8.1 \times 10^{-12}, \sigma = 1 \times 10^{-5}, \epsilon_s = 80.1, \epsilon_\infty = 5.5 \text{ in the material}$$

$$N = 1024, N_t = 12926$$

The sample rate for the data is one sample per .05ns

With this choice of parameters, the forward solve solution at $z = 0$ clearly shows distinct reflections from the z_1 , z_2 , and z_3 . This clear distinction will aid in our approximating the initial guesses, thus making this a relatively easy sample problem.

4.1.1 Initial Guesses

Assuming the physical parameters are given (either known or from a previous estimation), we want to determine the depth of any crack (d), and the width of that crack (δ), using reflection and/or transmission signals. First we will attempt to get very close approximations to d and δ using information about the travel times of the data signal, then we will use these values as initial guesses in the optimization routine.

We now describe how to identify the return time of each reflection, which in turn is used to estimate each z_i value, and therefore d and δ . We need to be able to identify the point in time when a reflection begins, i.e., the root immediately in front of a peak (or trough) in the signal. (Note that certain reflections are oriented peak first, while others are oriented trough first.) Three methods of doing this are as follows:

- i. $tr_1(i) = \min t$ such that $|\dot{E}(t)| > C_1$ and $t > C_2$, where C_1 is chosen to avoid noise being confused with reflections, and C_2 is large enough that the previous reflection has passed.
- ii. $tr_2(i) = \max t$ such that $|E(t)| < C_1$ and $t < C_2$, where C_1 is chosen to avoid noise being confused with reflections, and C_2 is small enough that the peak (or trough) has not yet passed.
- iii. $tr_3(i) = \max t_j$ such that $\hat{E}(t_j)\hat{E}(t_{j+1}) < 0$ and $t_j < C_2$, where C_2 is small enough that the peak (or trough) has not yet passed.

In our implementation we chose a combination of these methods in that we take the largest t that satisfies at least one of the above conditions. This seems to work better than any one method individually, especially in the presence of minor noise. Note also that certain reflections are oriented peak first, while others are trough first, thus in some cases C_2 is small enough that the trough has not yet passed.

We then use the difference between the return times of two reflections, multiplied by the speed of light in the medium (or vacuum, appropriately), to determine the additional distance travelled by the latter reflection. Namely, this is twice the distance between the two interfaces. In this manner we estimate d and, when possible, δ .

For small δ the two reflections are overlapping (see, for instance, Figure 8). In these cases we approximated the location of the start of the second reflection by the location of the trough. While this is a rough approximation, for this problem it was sufficient. In Problem 2 we will develop more sophisticated methods which could also be applied here.

4.1.2 Optimization of Least Squares Error

With these initial estimates to d and δ established, we now define our inverse problem to be: find $q = \{d, \delta\} \in Q_{ad}$ such that the following least squares error between the simulation and the observed data is minimized:

$$J(q) = \frac{1}{MS} \sum_{j=1}^M \sum_{i=1}^S (E(t_i, z_j^O; q) - \hat{E}_{ij})^2. \quad (6)$$

Here the \hat{E}_{ij} are measurements of the electric field taken at M specific locations (e.g. $z_1^O = 0$ and/or $z_2^O = 1$) and S distinct times (e.g., every $sr_e = 0.06ps$). The $E(t_i, z_j^O; q)$ are solutions of the simulations evaluated at the same locations and times corresponding to the data, \hat{E}_{ij} , and using parameter values q . The set Q_{ad} is the feasible set of q values determined such that d and δ are realistic (e.g. positive).

We apply an inexact Gauss-Newton iterative method to the optimization problem. That is, we re-write the objective function as

$$J(q) = \frac{1}{MS} R^T R$$

where $R_k = (E(t_i, z_j; q) - \hat{E}_{ij})$ for $k = i + (j - 1)M$ is the residual. To update our approximation to q we make the Inexact Newton update step $q_+ = q_c + s_c$ where

$$\begin{aligned} s_c &= - (R'(q_c)^T R'(q_c))^{-1} \nabla J(q_c) \\ &= - (R'(q_c)^T R'(q_c))^{-1} R'(q_c)^T R(q_c) \end{aligned}$$

is the step, q_c is the current approximation, and q_+ is the resulting approximation. This is an inexact method because we have disregarded the S Hessians of $(E(t_i, z_j; q) - \hat{E}_i)$, which is generally acceptable for small residual problems [7].

In this simple case we have a 2×2 matrix inverse, so we can compute it explicitly. Each iteration requires one function evaluation and a forward difference gradient, which is two additional function evaluations (since we have two parameters). Each function evaluation is equivalent to a simulation. Therefore we want as few iterations as possible.

4.1.3 Convergence

Initial testing (with $z = 0$ data only) shows convergence to 8 decimal places of each parameter in 6 iterations of Gauss-Newton with initial guesses having at most 5% relative error in δ and 2% in d . The algorithm does not converge to the correct solution if the initial guess for d has 5% relative error or if the initial guess for δ has 10% relative error.

One reason that the algorithm fails to converge is that this objective function is poorly behaved. In Figures 10 and 11 we show plots of the objective function with respect to δ . The two very large peaks in J on either side of the exact minimizer are due to the simulated solution going in and out of phase with the exact solution. For this example, the simulated solution is most out of phase with the exact solution at $\delta = .181$ and $\delta = .219$ which correspond to the first and second peak in J respectively. The signal received at $z = 0$ for these two δ values are plotted versus the actual signal with $\delta = .2$ in Figures 12 and 13. The same phenomenon occurs in the d direction, for the same reasons.

One may consider adding the information at the $z = 1$ signal to the least squares objective function hoping to lessen the effects of the peaks in J . But as demonstrated in Figures 14 and 15, which show the objective function using both signals and using just the $z = 1$ signal respectively, the out of phase phenomena is still present.

Very few optimization routines can expect global convergence without having initial conditions sufficiently in between the two peaks in J . The effective convergence region for this objective function applied to this problem (with or without observations at $z = 1$) is within about 8% of the actual value of δ when d is exact and within about 7.5% of the actual value of d when δ is exact.

Note also that the convergence region is very dependent on the frequency of the interrogating signal; for higher frequencies, the region is even smaller. This is because the distance between the two peaks in J is

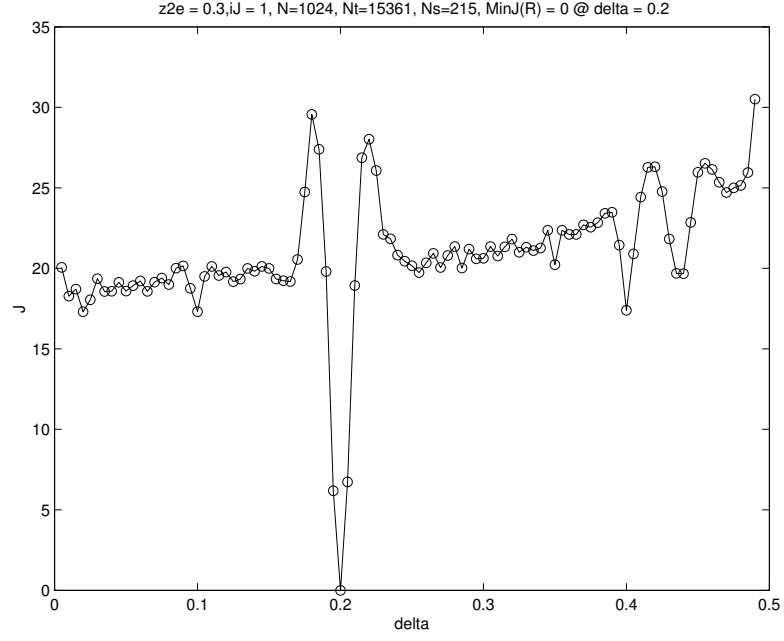


Figure 10: Nonlinear Least Squares objective function versus δ for a large range of δ values (data at $z = 0$ only)

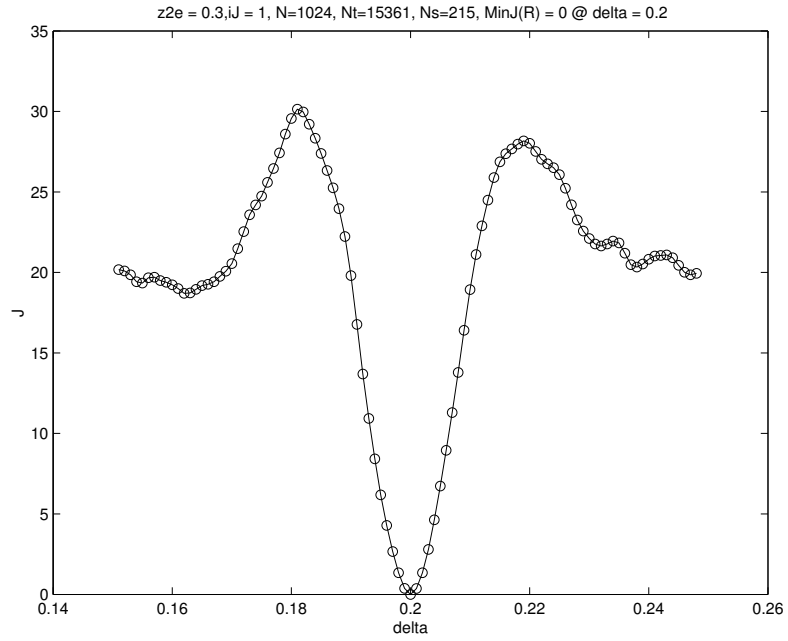


Figure 11: Nonlinear Least Squares objective function versus δ for a small range of δ values (data at $z = 0$ only)

linearly dependent on λ , the wave length of the interrogating signal. This has profound implications for our desire to interrogate with signals in the terahertz range.

Further, this is considering only a one parameter minimization problem with the other parameter held fixed at the exact solution. Convergence is much worse for the actual two parameter problem. The difficulties

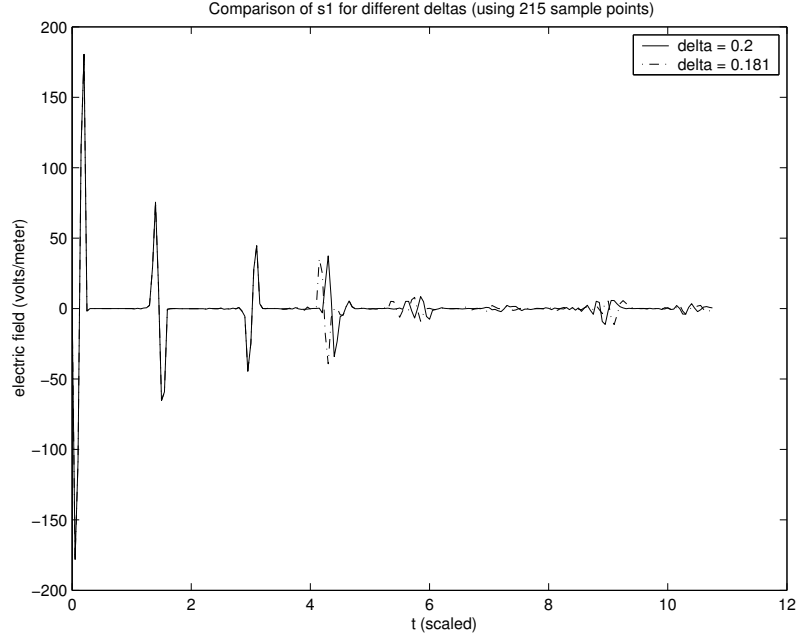


Figure 12: Signal received at $z = 0$ for two different values of δ demonstrating the simulation going out of phase on the left with the data signal

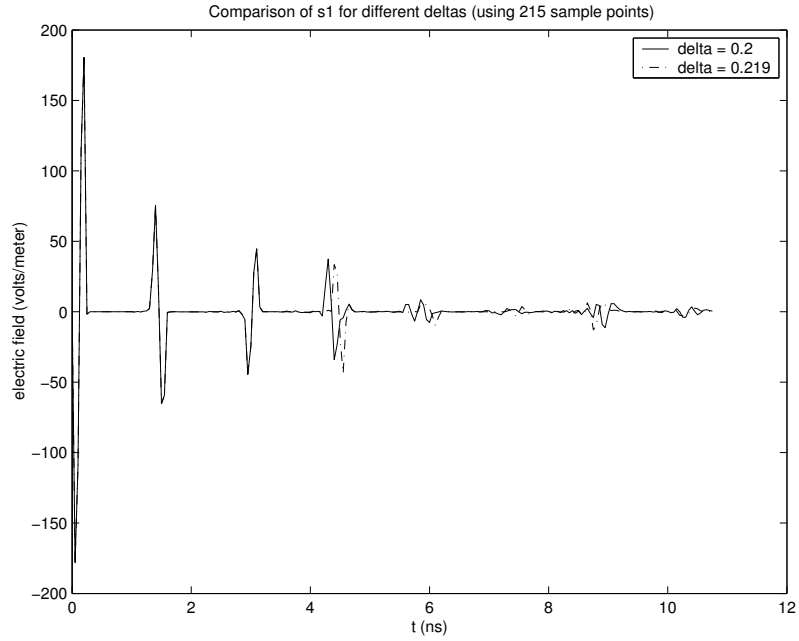


Figure 13: Signal received at $z = 0$ for two different values of δ demonstrating the simulation going out of phase on the right with the data signal

with the least squares objective function in this problem are clearly visible in the surface plots shown in Figures 16 and 17. This explains why d has a smaller convergence region, since J is much more sensitive to d . Also we see that there are peaks in J in the d direction as well.

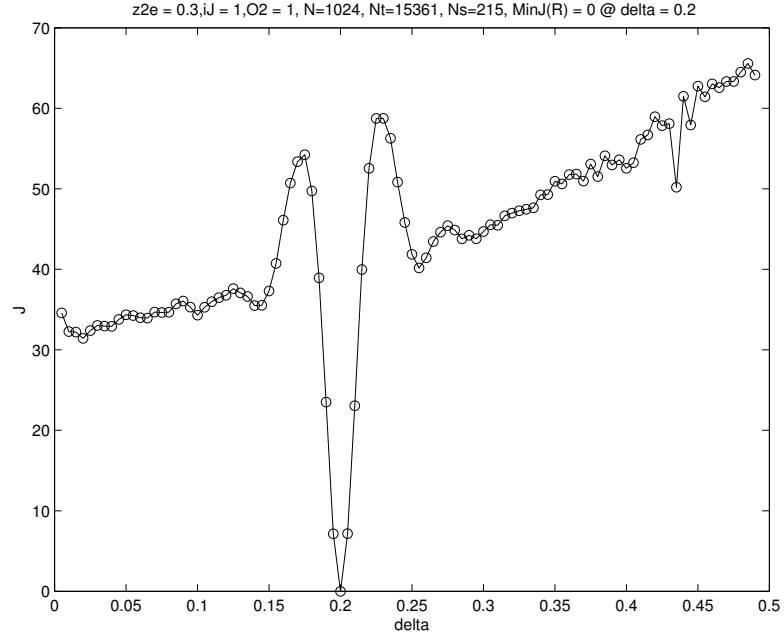


Figure 14: Nonlinear Least Squares objective function, using signals received at both $z = 0$ and $z = 1$, versus δ

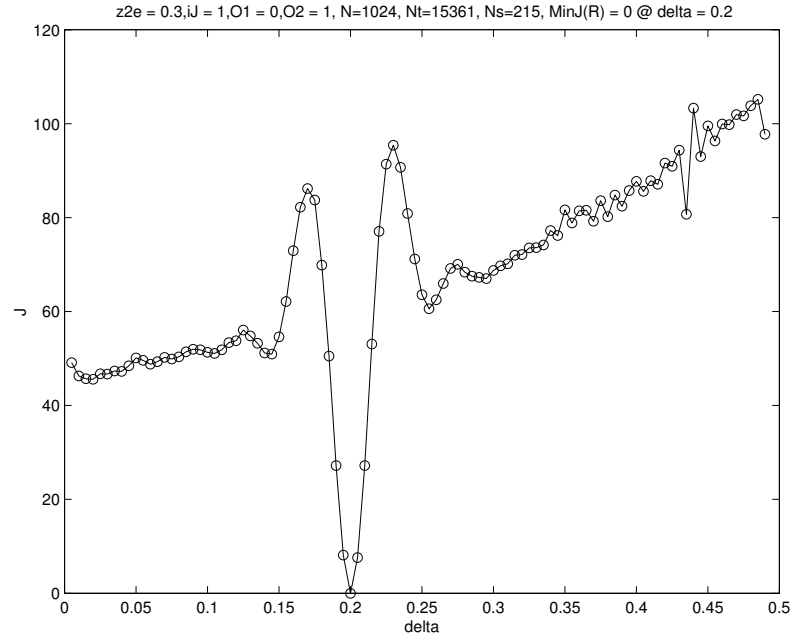


Figure 15: Nonlinear Least Squares objective function, using signals received at $z = 1$ only, versus δ

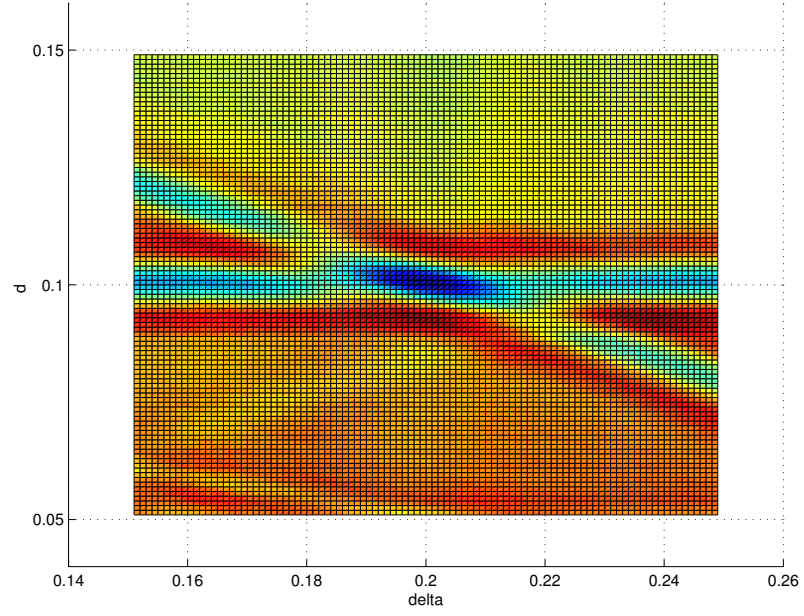


Figure 16: Nonlinear Least Squares objective function (J) versus δ and d projected to $2D$ (data at $z = 0$ only)

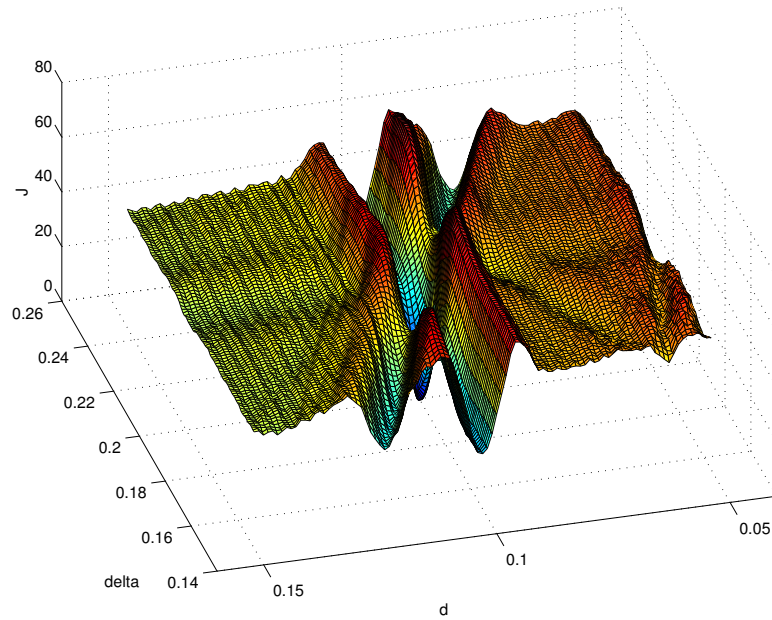


Figure 17: Nonlinear Least Squares objective function (J) versus δ and d in $3D$ (data at $z = 0$ only)

The diagonal “trench” occurs approximately along the line

$$d = -\frac{1}{2}(\delta - \delta^*) + d^* \quad (7)$$

(where $*$ denotes the exact solution). This happens because when the depth d is decreased by, say a , the z_1 and z_2 reflections of the simulation no longer correspond with those of the data (think of the two overlapping reflections at $t = 1.5ss$ in Figure 8 being shifted slightly to the left in some simulation).

With d decreased by a we must increase δ to try to compensate and get as much of the signal from the simulation to correspond with the data. Since δ only affects the z_2 reflection, it is clear that the z_1 reflections will not correspond. If δ is increased so that the distance traveled by the z_2 reflection is increased by b , then we will need for the time required for a signal to travel a distance a in the material to be equal to the time required for a signal to travel a distance b in a vacuum. Since the distance traveled by the z_2 reflection is twice that of the increase in δ , we compute the increase in δ to be $\sqrt{\epsilon_\infty}a$. In our case $\epsilon_\infty = 5.5$, so the increase is $2.3452a$. This explains the slope of approximately $-\frac{1}{2}$ in the relationship (7).

4.2 An Improved Objective Function

As demonstrated above, the usual Nonlinear Least Squares objective function has two large peaks in J on either side of the exact minimizer. The reason for these peaks in J is that the simulated solution goes in and out of phase with the data as d or δ change. When they are precisely out of phase, there is a very large absolute error, which when squared, causes the objective function to have large peaks. Therefore, a logical solution would be not to consider the absolute error, but the error of the absolute values, i.e., the following objective function:

$$J_2(q) = \frac{1}{2S} \sum_{i=1}^S \left(|E(t_i, z_i^O; q)| - |\hat{E}_i| \right)^2. \quad (8)$$

This will prevent the fact that the signals go in and out of phase with each other from having an impact on the objective function, since positive magnitudes cannot cancel each other out. This gives a more accurate measure of the difference between two signals. Note that $J_2(q)$ is not differentiable on a set of measure zero; this is very unlikely to affect the finite difference computations of the gradients, and did not seem to present problems in our testing. We plot $J_2(q)$ versus δ in Figures 18 and 19.

Note that while we have effectively eliminated the peaks on either side of the exact solutions, in essence we have merely converted them to local minima! But, since the minima occur for the same reasons the peaks in J had been occurring, they occur at the same values of δ . Note that we can see from plotting the signals that they were exactly out of phase when they were shifted by $\frac{\lambda}{4}$, where λ is the wavelength of the interrogating signal. Therefore δ is off by $\frac{\lambda}{4}$. Since we determine the frequency of the interrogating signal, this is a known quantity, and we can predict where these local minima will occur a priori!

Most optimization routines will continue until they find a local minimum, and since the two false minima described above are at least close to “predictable” locations, we can easily test on either side of any detected minimum to determine if it is in fact global. If J_2 is less at a fixed distance in the δ direction (e.g., $\frac{\lambda}{4}$) on either side of a detected minimum, i.e., at either *test location* of the local minimum, we restart Gauss-Newton at the new best guess. Thus if either of the two local minima described above is found, we will eventually have global convergence if one of their *test locations* are sufficiently close to the global minimizer.

To graphically demonstrate this approach, we have added dotted and dash-dotted lines to the the graph of $J_2(\delta)$ in Figure 19 to represent the test values of the first and second local minima respectively. One can clearly see that if either local minimum is detected, the test value either to its right if it is the first one, or to its left if it is the second, will give a smaller J_2 and should eventually lead to the global minimizer being detected. Using this method we have in principle increased our convergence region to about 25% of δ when d is exact.

Our objective function also has local minima in the d direction, as demonstrated in Figure 20. Again they are in predictable locations, being at roughly $d^* \pm \frac{\lambda}{4\sqrt{\epsilon_\infty}}$, where d^* represents the global minimizer. Thus, if we know λ and ϵ_∞ (which we should), then we can determine exactly where to test for the global minima once a local minimum has been found. Using the above method, we have in principle increased the d convergence region from about 7.5% to about 15% when δ is exact.

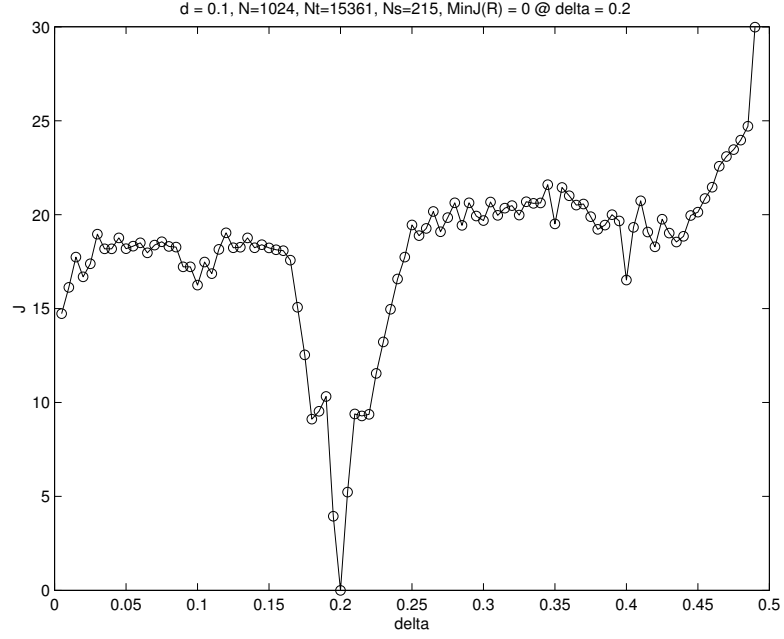


Figure 18: Our modified Nonlinear Least Squares objective function (J_2) versus δ for a large range of δ values

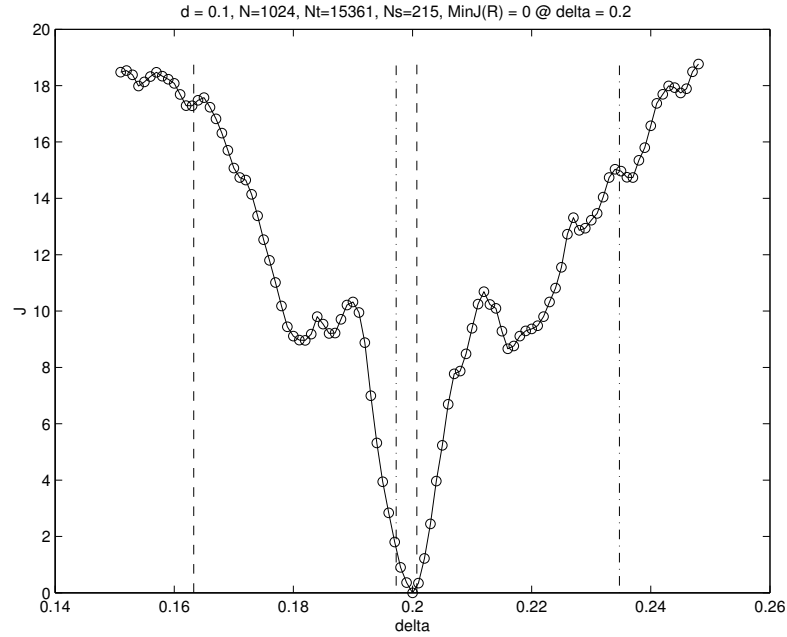


Figure 19: Our modified Nonlinear Least Squares objective function (J_2) versus δ for a small range of δ values. The dotted lines represent the delta values that will be tested if a local minimum is found

Other possible modifications to the Least Squares objective function having similar effects include squaring the signal instead of taking the absolute value (thus preserving smoothness everywhere), or just halving the reference signal so that we only have a positive amplitude to begin with.

In addition to the two new minima created above, we still have other local minima to deal with which are present in both formulations of the objective function. Some of these local minima can be reduced by

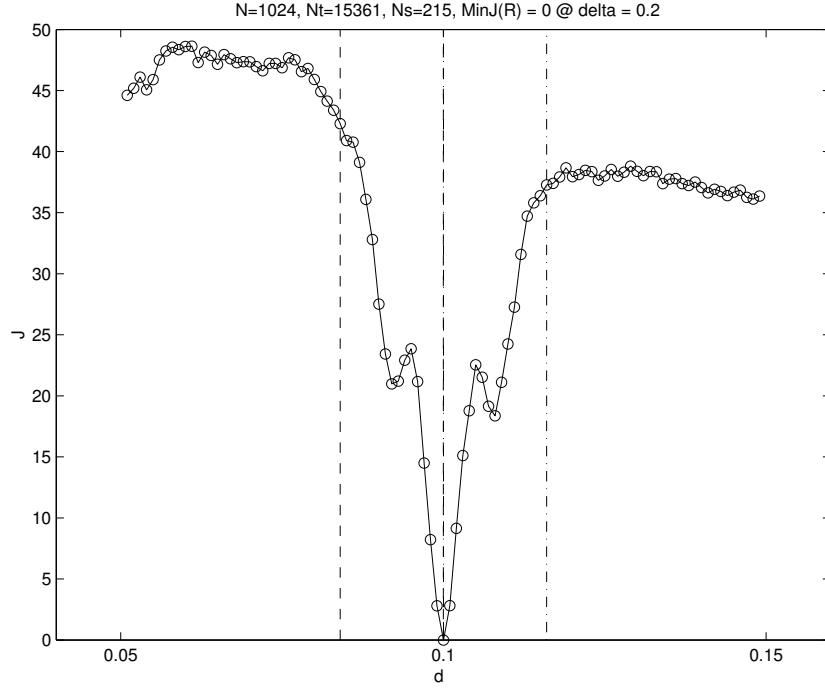


Figure 20: Our modified Nonlinear Least Squares objective function (J_2) versus d for a small range of d values. The dotted lines represent the delta values that will be tested if a local minimum is found

using a smaller mesh size (more finite elements) since they are the result of numerical error (see Figures 21 and 22).

All of the above discussions and Figures 18 through 22 involve J_2 with observations only at $z = 0$. We have also done tests using only the $z = 1$ transmitted data in the objective function from (8) (see Figure 23). Clearly this function does not have the same problem of the third reflections (since $z = 1$ only sees the even numbered reflections). But it does still have a very deep and significantly close local minima of its own. This minimum is caused by the same type of problem that causes the $z = 0$ minimum. However, we have found that by combining both $z = 0$ and $z = 1$ data in the objective function J_2 we can get much better results since their deficiencies somewhat average out (see Figure 24).

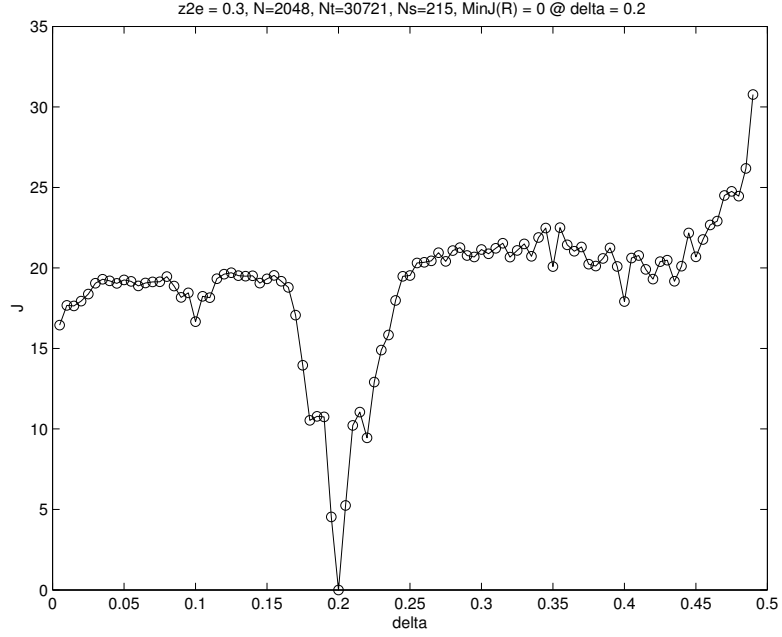


Figure 21: Our modified Nonlinear Least Squares objective function (J_2), using a simulation with twice as many meshes, versus δ for a large range of δ values

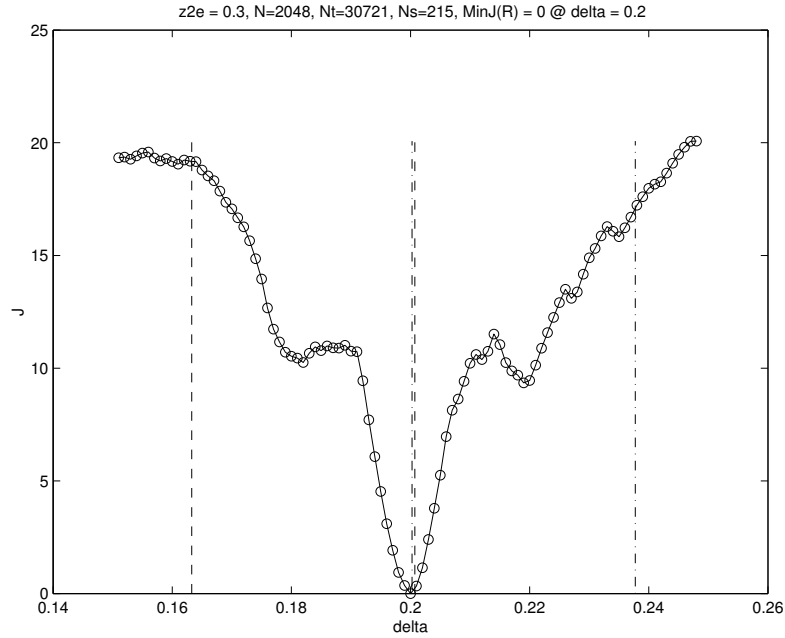


Figure 22: Our modified Nonlinear Least Squares objective function (J_2), using a simulation with twice as many meshes, versus δ for a small range of δ values. The dotted lines represent the delta values that will be tested if a local minimum is found

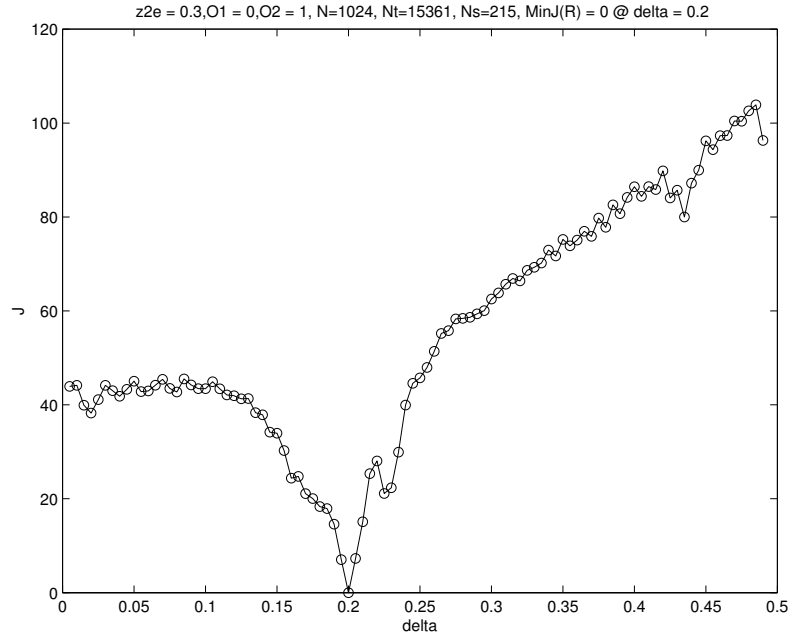


Figure 23: Our modified Nonlinear Least Squares objective function (J_2), using the signal received at $z = 1$ only, versus δ for a large range of δ values

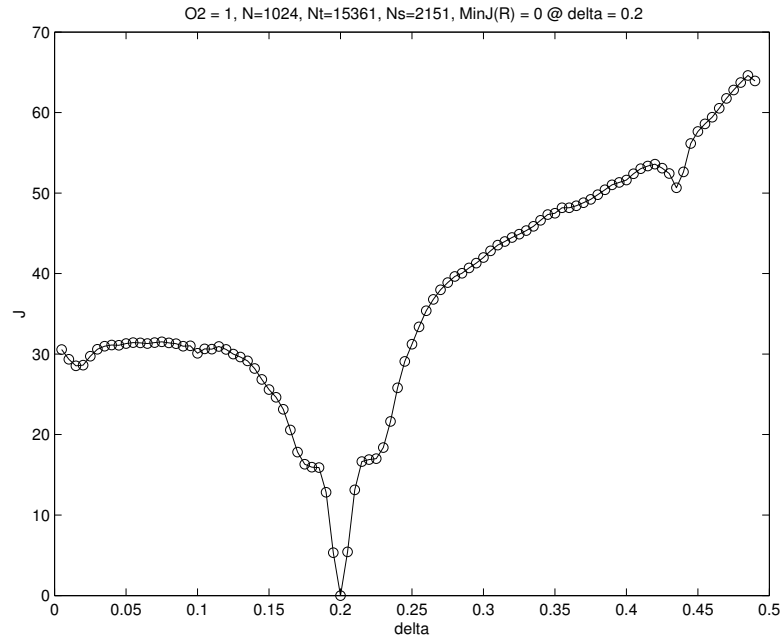


Figure 24: Our modified Nonlinear Least Squares objective function (J_2), using the signal received at $z = 0$ and $z = 1$, versus δ for a large range of δ values

4.3 Testing J_2

In order to determine the limitations of an optimization routine to minimize our objective function J_2 in a more practical setting we examine J_2 versus q when error is present. In particular we try both adding random noise to the data signal, as well as testing bad initial guesses for δ and d . It should be noted that although we have previously established that there are certain benefits to having data at $z = 1$ in the tests reported on below we assume that it is not available and used only observations at $z = 0$.

4.3.1 Sensitivity to Initial Guesses

For J_2 described in (8) we compute a surface plot for various δ and d values, which is shown in Figures 25 and 26. We notice immediately that the objective function is much more sensitive to d than to δ , therefore it is imperative that our initial guess for d is as good as possible.

To give an idea of what may happen if our d guess were not within the 15% our testing has determined is necessary, we plot the objective function versus δ for three values of d , which are 3%, 15%, and 30% off respectively, in Figures 27, 28, and 29.

It is clear that the 15% case should have no trouble converging with relatively good initial guess for δ , but even if the initial value for δ were exact, the 30% case could quite possibly converge to the minimum at the far left of Figure 29. It may be surprising that there should be a minimum for very small δ values at all, even more so that it is in fact the global minimum! This can be explained by first remembering that, for example in the 15% case, the original data was computed with $d = .1$ but the simulations used $d = .085$, so we should expect that regardless of δ there is a part of the simulated signal that will not match the data, namely the first reflection. Thus the first reflection of the data is not matched by the simulation, unless δ is small enough to match it, e.g. $\delta = .035$, and this is exactly what is happening, as displayed in Figure 30. Figure 31 shows the other local minimum of the 15% case where the z_3 reflection of the simulation does match the z_3 reflection of the data.

Notice that the distance between the two minimum values of δ (see for example Figure 28 or 29) is exactly $\delta^* = 0.2$, which is what would be expected. However, we cannot apply the same idea as before where we add or subtract a fixed amount to test for other local minima, since for one, the “more optimal” of the two is farther from the “true” solution, and also, we would have to know δ in the first place to add or subtract it, but that is what we are trying to estimate!

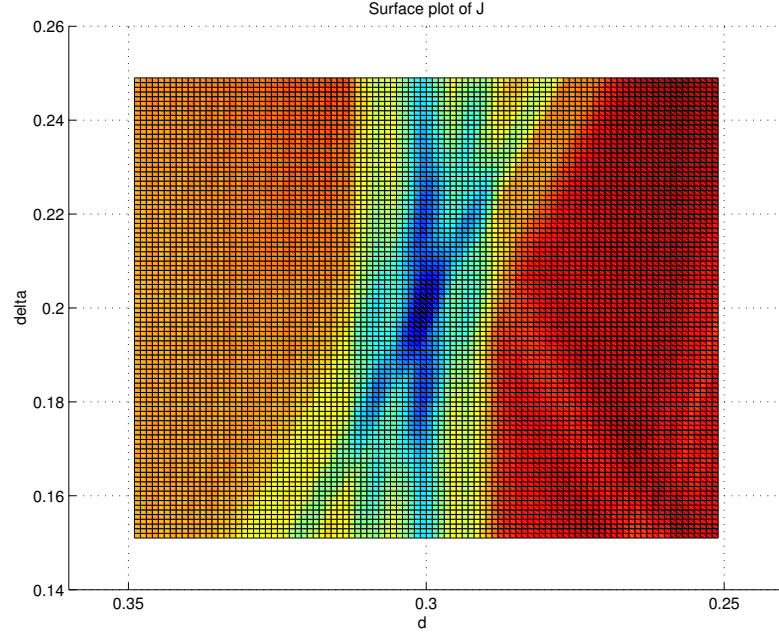


Figure 25: Our modified Nonlinear Least Squares objective function (J_2) versus δ and d projected to $2D$

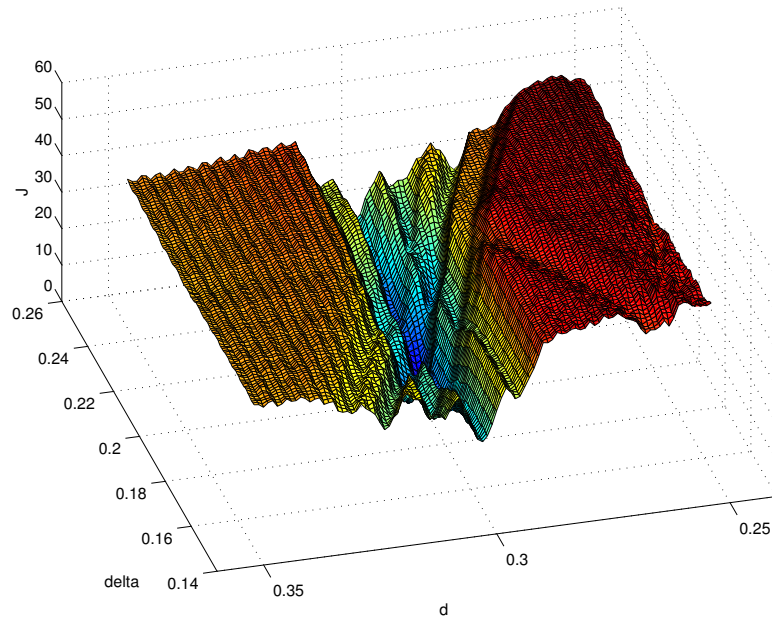


Figure 26: Our modified Nonlinear Least Squares objective function (J_2) versus δ and d in $3D$

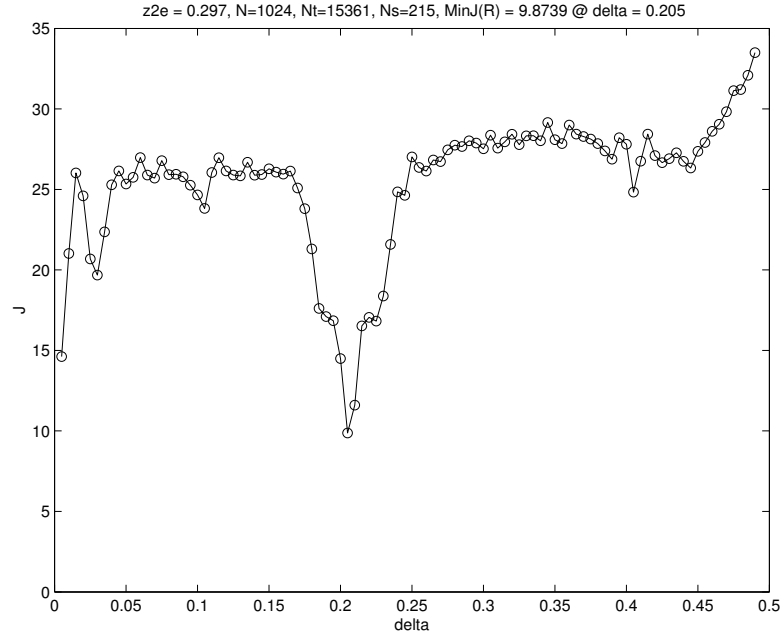


Figure 27: Our modified Nonlinear Least Squares objective function (J_2) versus δ with $d = .097$ (representing 3% relative error)

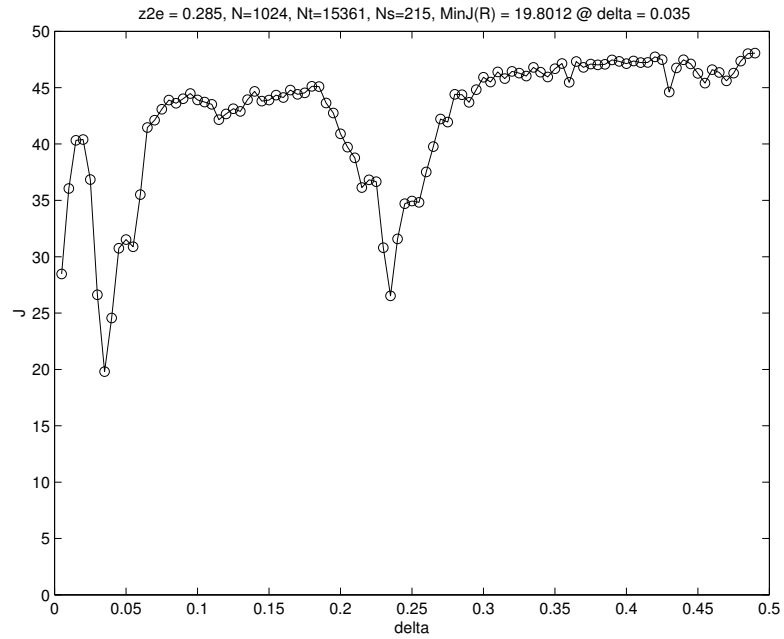


Figure 28: Our modified Nonlinear Least Squares objective function (J_2) versus δ with $d = .085$ (representing 15% relative error)

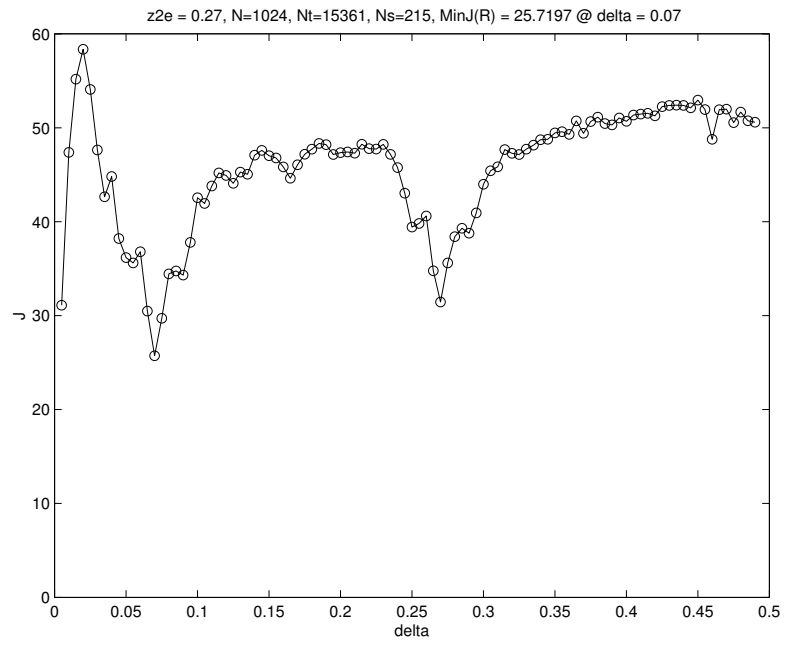


Figure 29: Our modified Nonlinear Least Squares objective function (J_2) versus δ with $d = .07$ (representing 30% relative error)

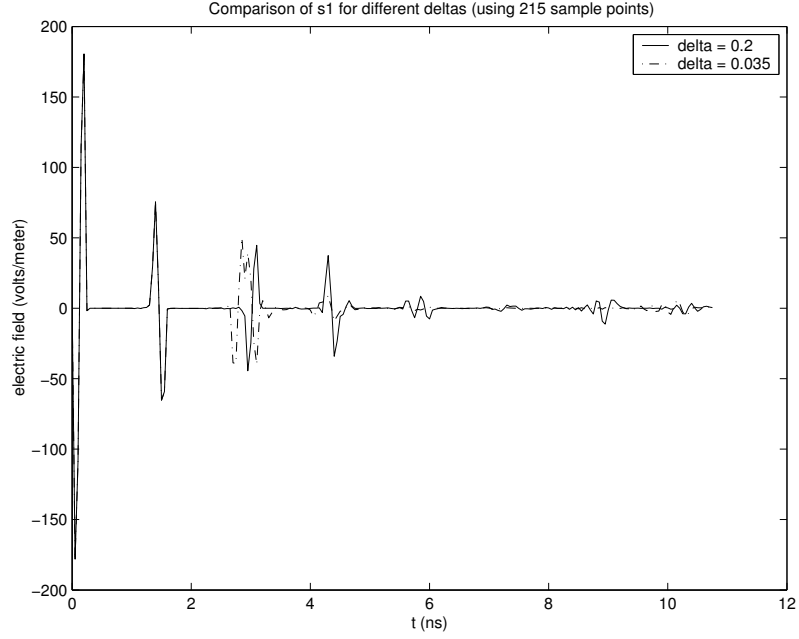


Figure 30: Signals received at $z = 0$ (where $(d, \delta) = (.1, .2)$ corresponds to the given data and $(d, \delta) = (.085, .035)$ corresponds to the simulation) demonstrating the z_2 reflection of the simulation corresponding to the z_1 reflection of the data

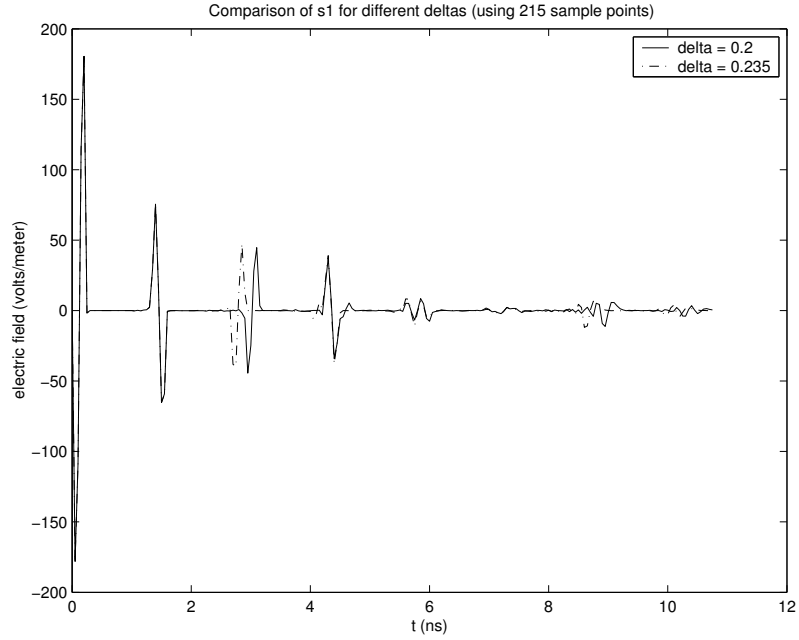


Figure 31: Signals received at $z = 0$ (where $(d, \delta) = (.1, .2)$ corresponds to the given data and $(d, \delta) = (.085, .235)$ corresponds to the simulation) demonstrating the z_2 reflection of the simulation matching the z_2 reflection of the data and ignoring the z_1 reflection

4.3.2 Random Observation Noise

In order to test the feasibility of this procedure as an estimation method, we have produced synthetic data for our observations \hat{E}_i . In an actual experiment, one must assume that the measurements are not exact. To simulate this we have added random noise to the original signal. The absolute value of the noise is relative to the size of the signal. If E_i is the data sampled, then we define $\hat{E}_i = E_i(1 + \nu\eta_i)$, where η_i are independent normally distributed random variables with mean zero and variance one. The coefficient ν determines the relative magnitude of the noise as a percentage of the magnitude of E_i , in particular, $\nu = 0.05$ corresponds to 10% noise and $\nu = 0.025$ to 5% noise.

Plots of the resulting objective functions for various values of ν ranging from 2% to 40% are shown in Figures 32-37. We notice that the structure of the curves is not significantly affected, nor is the location of the global minimum. However the magnitude of the objective function is increased, making Inexact Newton methods slightly less reliable due to the larger residual. Still, our results show that the minima were consistently found and within a reasonable amount of time. Select examples are summarized in Table 1.

ν	d	δ	J	Iterations	CPU Time (s)
0	0.1	0.2	1.32319E-10	7	160
0.01	0.099994	0.199969	0.00792792	8	186
0.05	0.099974	0.199835	0.199489	13	291
0.2	0.099928	0.199204	3.04619	20	435

Table 1: Number of Iterations and CPU Time for Gauss-Newton given various relative magnitudes of random error

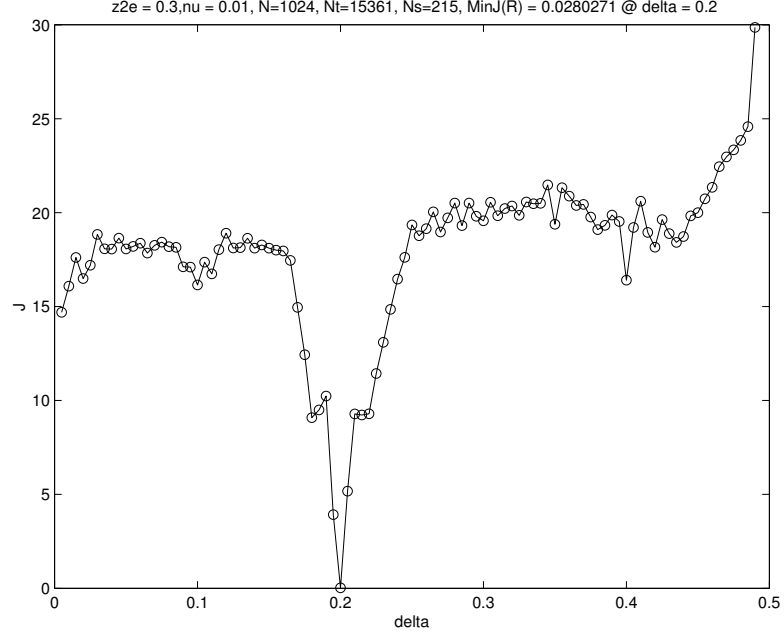


Figure 32: Our modified Nonlinear Least Squares objective function (J_2), with 2% random noise, versus δ for a large range of δ values

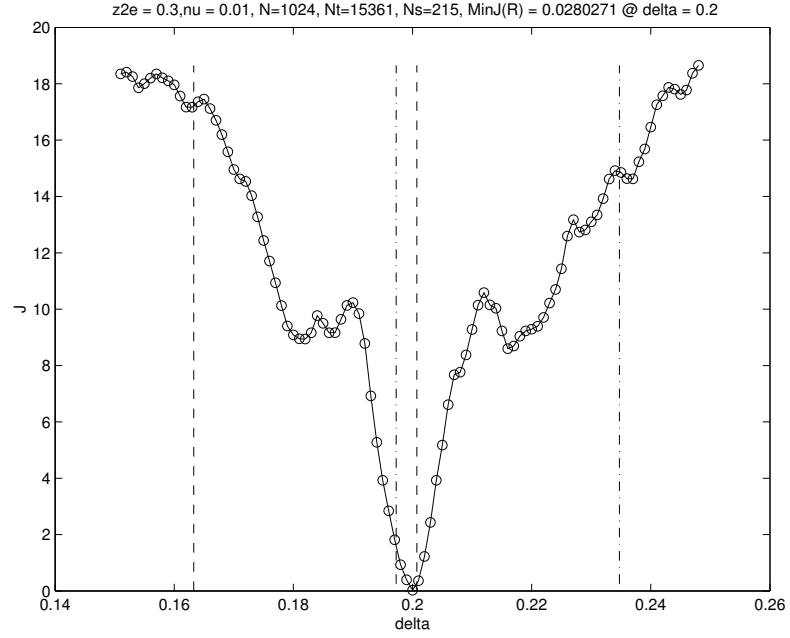


Figure 33: Our modified Nonlinear Least Squares objective function (J_2), with 2% random noise, versus δ for a small range of δ values

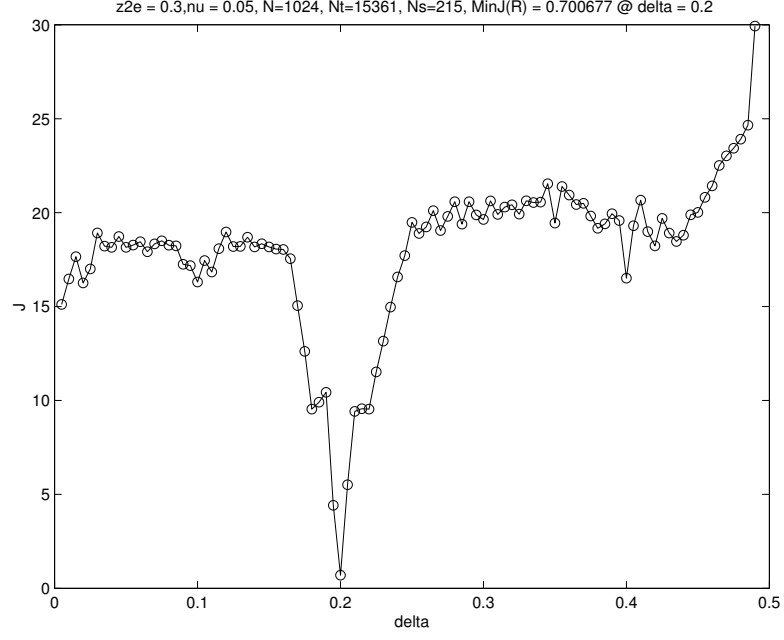


Figure 34: Our modified Nonlinear Least Squares objective function (J_2), with 10% random noise, versus δ for a large range of δ values

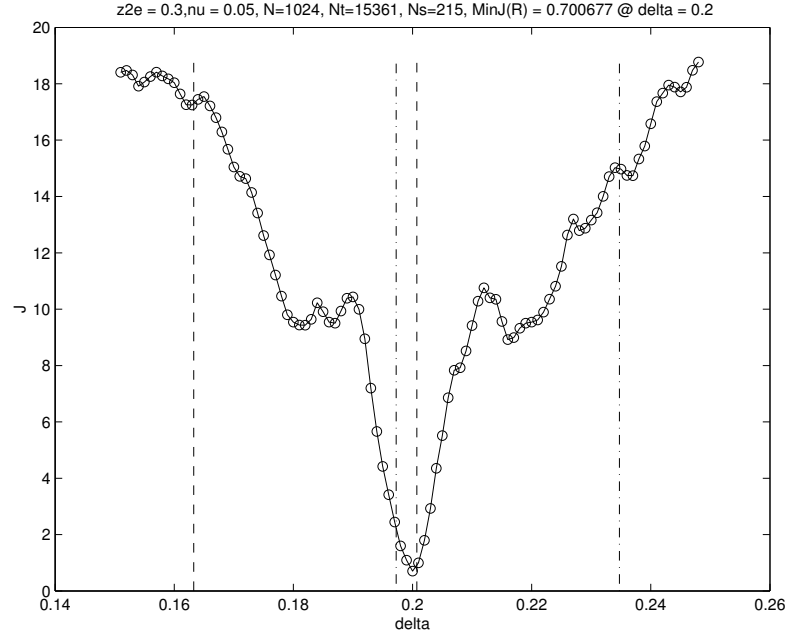


Figure 35: Our modified Nonlinear Least Squares objective function (J_2), with 10% random noise, versus δ for a small range of δ values

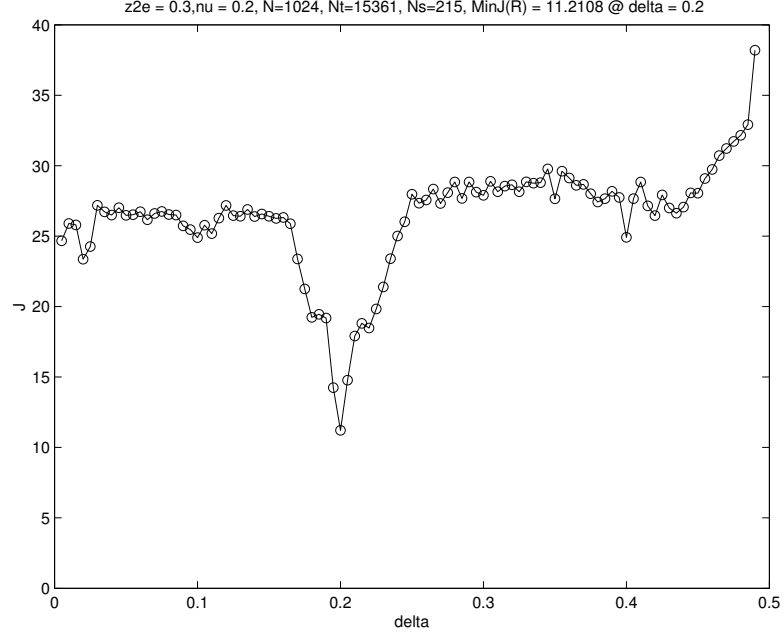


Figure 36: Our modified Nonlinear Least Squares objective function (J_2), with 40% random noise, versus δ for a large range of δ values

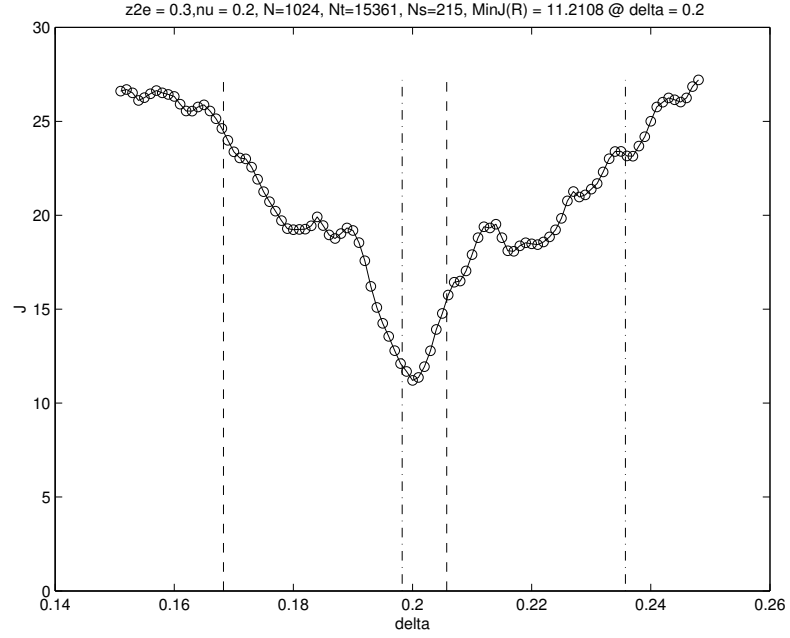


Figure 37: Our modified Nonlinear Least Squares objective function (J_2), with 40% random noise, versus δ for a small range of δ values

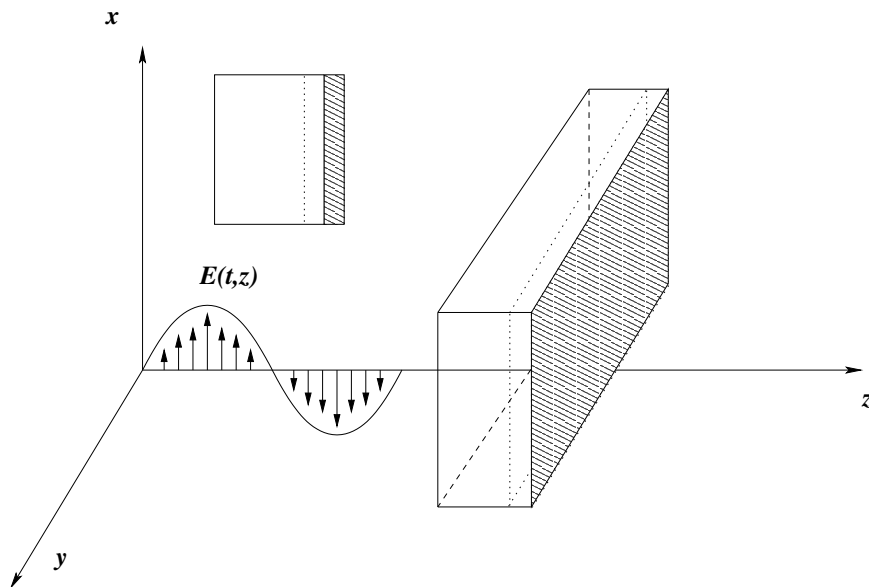


Figure 38: Schematic of Problem 2: determining the depth and width of a gap between a dielectric slab and a metallic backing

5 Problem 2

We will apply the most useful techniques from Problem 1 to a new formulation of the interrogation problem. In Problem 2 we consider a dielectric slab and a metallic backing (conductor) with a possible gap between the two (see Figures 38 and 39). Applications of this specific formulation included detecting delamination of insulation from metallic containers, e.g., insulating foam on a space shuttle fuel tank. In order for this numerical approach to be useful in this particular application we must be able to resolve a slab thickness of at least $20cm$ and a frequency of $100GHz$.

We will again assume the same physical parameters for our dielectric and consider the gap as a vacuum. The variables d and δ are still the depth and the width of the gap respectively. One major difference is that in this problem we are only able to detect the electromagnetic signal in front of the material. Also, since the metallic backing reflects much of the signal, we have considerably more overlapping of the reflections to worry about. These properties contribute to the fact that this formulation leads to a much more difficult inverse problem. For this reason we will be using more sophisticated optimization routines including a Levenberg-Marquardt parameter and Implicit Filtering. We will also need to develop different approximation methods for our initial guesses.

The implementation of this problem has several minor differences from the previous one. First, we now only need to represent two interfaces \tilde{z}_1 and \tilde{z}_2 , with \tilde{z}_0 and \tilde{z}_3 being the front and back boundaries, respectively. Thus now we define the depth of the gap as $d := \tilde{z}_2 - \tilde{z}_1$ and the width as $\delta := \tilde{z}_3 - \tilde{z}_2$. Also, as previously mentioned, the conductive metal backing reflects the signal, and hence we must change our absorbing boundary conditions at $z = 1$ (for a finite computational domain), to an actual fixed, Dirichlet boundary condition ($E = 0$). We must modify our finite element matrices accordingly, as well. Otherwise, the numerical method for simulation is the same as it was for Problem 1, namely standard finite element methods for spatial derivatives, and an alternating implicit/explicit centered difference time stepping scheme. Sample solutions are plotted in Figure 40.

We again define our inverse problem to be: find $q := \{d, \delta\} \in Q_{ad}$ such that an objective function representing the error between the simulation and the observed data is minimized:

$$\min_{q \in Q_{ad}} J(q).$$

Here the measurements of the electric field, \hat{E}_i , are taken only at $z = 0$, but still at S distinct times (e.g.,

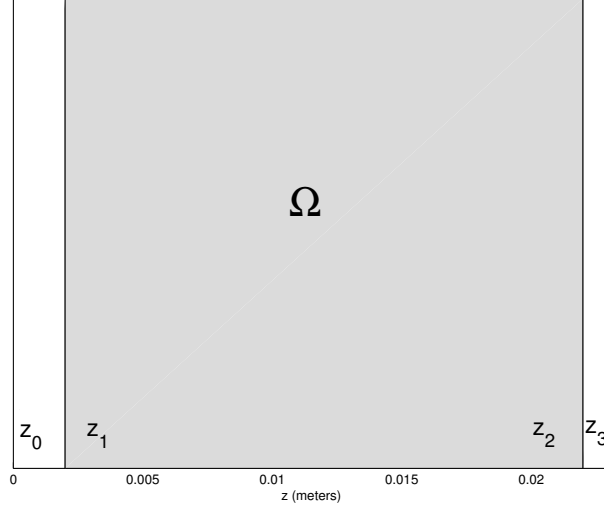


Figure 39: The domain of the material slab with a gap between the medium and a metallic conductive backing: $\Omega = \{z|z_1 \leq z \leq z_2\}$.

every $0.06ps$). The solutions of the simulations, $E(t_i, 0; q)$, are evaluated at the same location and times corresponding to the given data, and using parameter values q . In lieu of actual data from experiments, we again create our observed data by using the simulator, however, the only information that is given to the minimizer is the data observed at $z = 0$, which we will denote by \hat{E} .

The system that we use to represent the propagation of the electric field, and thus simulate in order to solve our inverse problem, is as follows, and includes the above mentioned Dirichlet condition at $z = 1$:

$$\begin{aligned}
\mu_0\epsilon_0(1 + (\epsilon_\infty - 1)I_\Omega)\ddot{E} + \mu_0I_\Omega\ddot{P} + \mu_0\sigma\dot{E} - E'' &= -\mu_0\dot{J}_s && \text{in } \Omega \cup \Omega_0 \\
\tau\dot{P} + P &= \tau\epsilon_0(\epsilon_s - \epsilon_\infty)E && \text{in } \Omega \\
[\dot{E} - cE']_{z=0} &= 0 \\
[E]_{z=1} &= 0 \\
E(0, z) &= 0 \\
\dot{E}(0, z) &= 0
\end{aligned}$$

with

$$J_s(t, z) = \delta(z)\sin(\omega t)I_{[0, t_f]}(t).$$

See Section 2 for a complete description.

5.1 Objective Function

As in the previous problem, we encounter difficulties when attempting to use the standard Least Squares objective function to compute the error between the simulated signal and the observed data. The constructive interference of peaks and troughs produces peaks in J in the objective function on all sides of the global minimum which make it nearly impossible to find the solution in the middle. The peaks in J in the d direction are clearly apparent in Figure 41. In contrast, Figure 42 shows a surface plot of our modified least squares objective function

$$J_2(q) = \frac{1}{2S} \sum_{i=1}^S \left| |E(t_i, 0; q)| - |\hat{E}_i| \right|^2.$$

Remark 3 *The gradual slope down and to the left in the surface plots is due to the increased attenuation of signals as the depth is increased. Also, the drop off at large depth values is an artifact of having a finite end time for collecting data, i.e., the reflection from the back of the material did not return in time.*

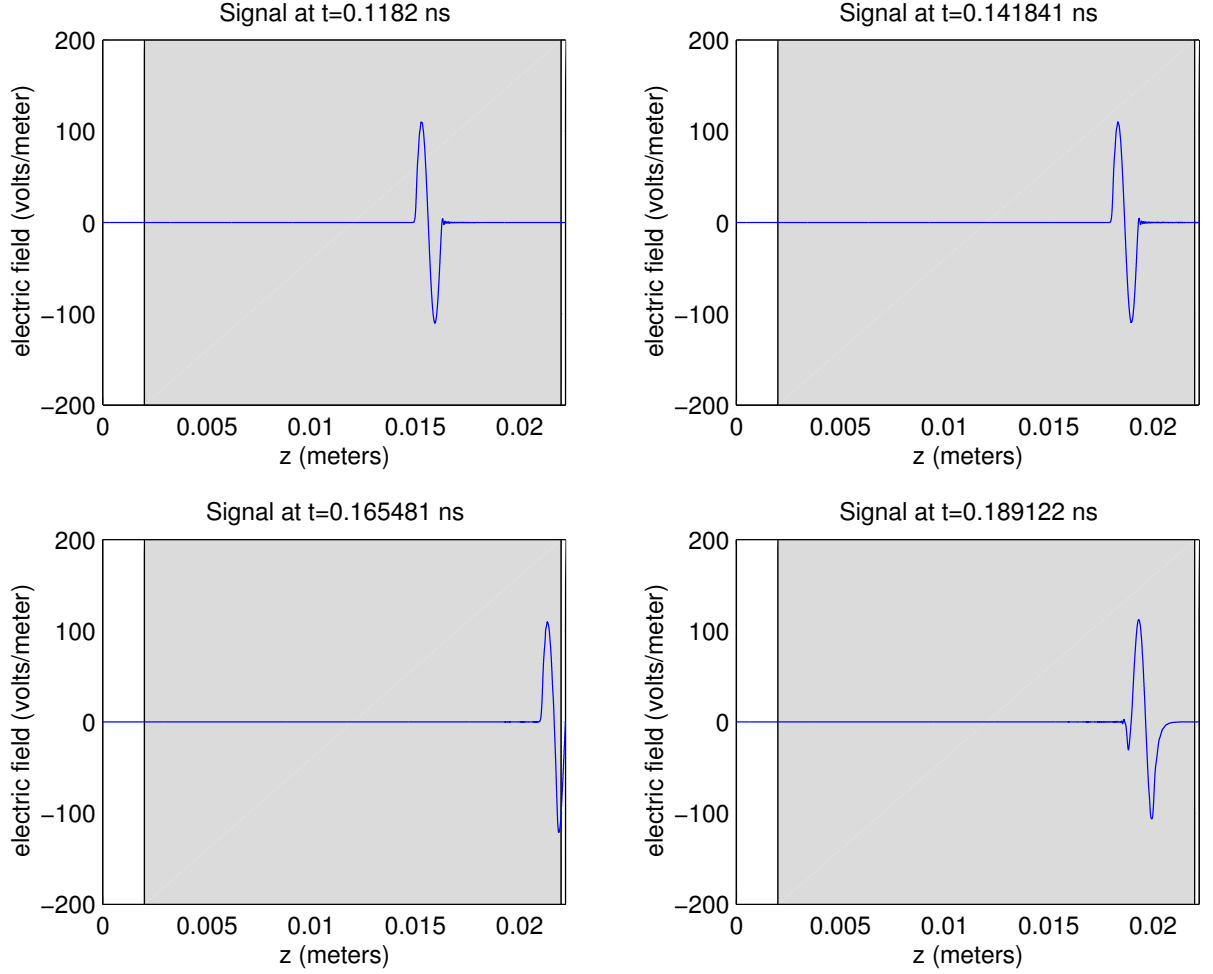


Figure 40: Computed solutions at different times of a windowed electromagnetic pulse incident on a Debye medium with a crack between the medium and a metallic conductive backing. The width of the slab is $d = .02m$ and the width of the gap is $\delta = .0002$ (barely visible at the far right of the gray region).

It is clear, as before, that the initial guess is crucial to the success of any optimization routine. To further demonstrate this importance, we expand our surface plot out in the δ direction, and zoom in with respect to d . Figures 43 and 44 show J_1 and J_2 respectively. Notice that although J_2 does not exhibit the familiar peaks in J of J_1 , it does however still have many local minima, which are as difficult, if not more so, to avoid in a minimization routine.

The local minima occur approximately every $\frac{\lambda}{4}$ along the line

$$d = -\frac{1}{\sqrt{\epsilon_\infty}}\delta + b.$$

This happens for the same reason as in Problem 1 (see Section 4.1.3). For an illustration, see Figures 45 and 46. In each figure we see that the small objective function value is due primarily to the coinciding of the single largest peak and/or the largest trough. (Note that the trailing oscillations are not numerical noise; the reflections appear as they do because $\delta \approx \frac{\lambda}{4} \approx 7.5 \times 10^{-4}$.)

Because we cannot eliminate these local minima, we must appeal to the procedure that worked in the previous problem, namely testing “check points”. Since we know where these local minima are occurring with respect to the global minimum, if our minimization routine finds what it suspects to be a local minima,

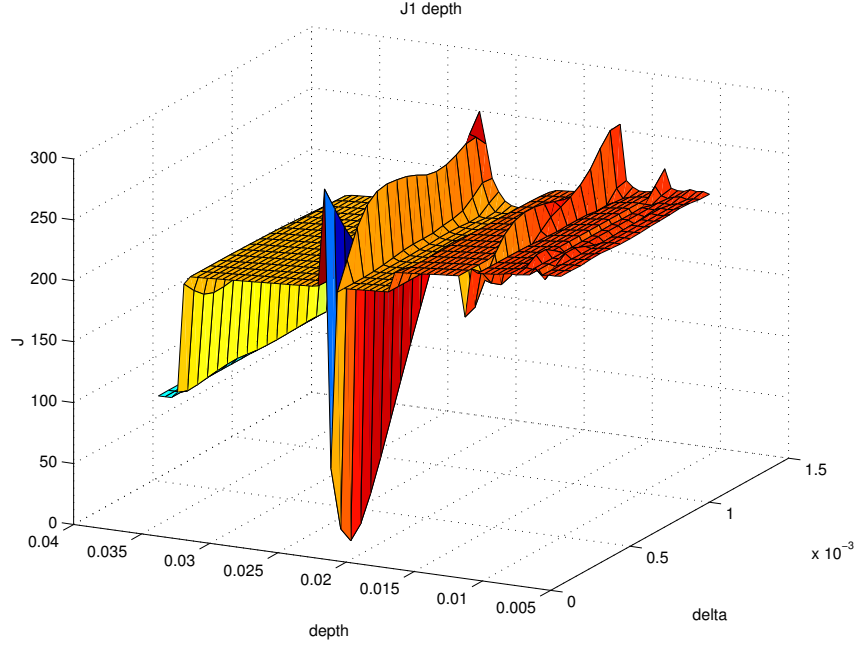


Figure 41: Surface plot of Least Squares objective function demonstrating peaks in J .

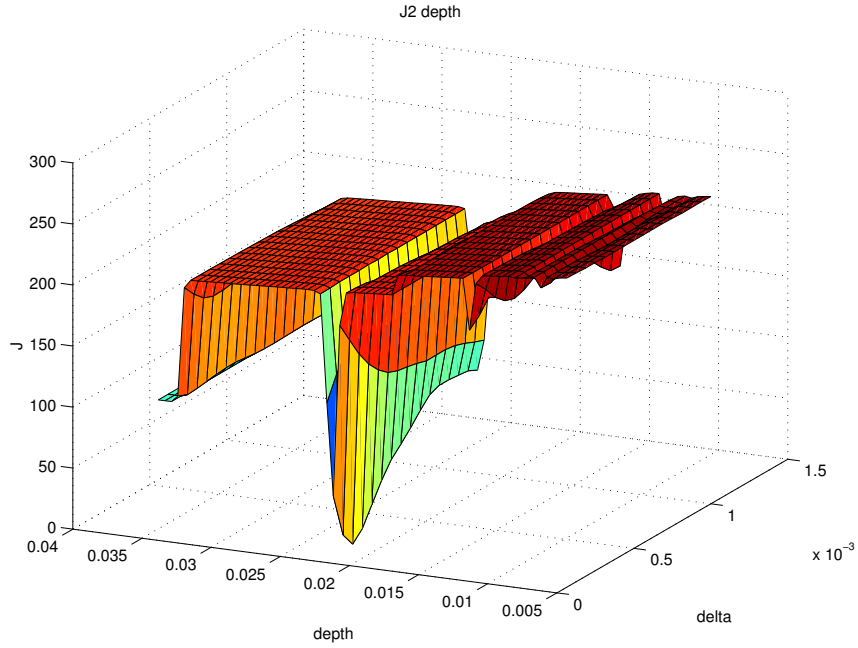


Figure 42: Surface plot of modified Least Squares objective function demonstrating lack of peaks in J .

say (d_1, δ_1) , we simply check $(d_1 \pm \alpha \frac{\lambda}{4}, \delta_1 \mp \alpha \sqrt{\epsilon_\infty} \frac{\lambda}{4})$, where $\alpha = 1/\sqrt{1 + \epsilon_\infty}$. If we find a lower objective function value, we restart our optimization routine at that “check point”.

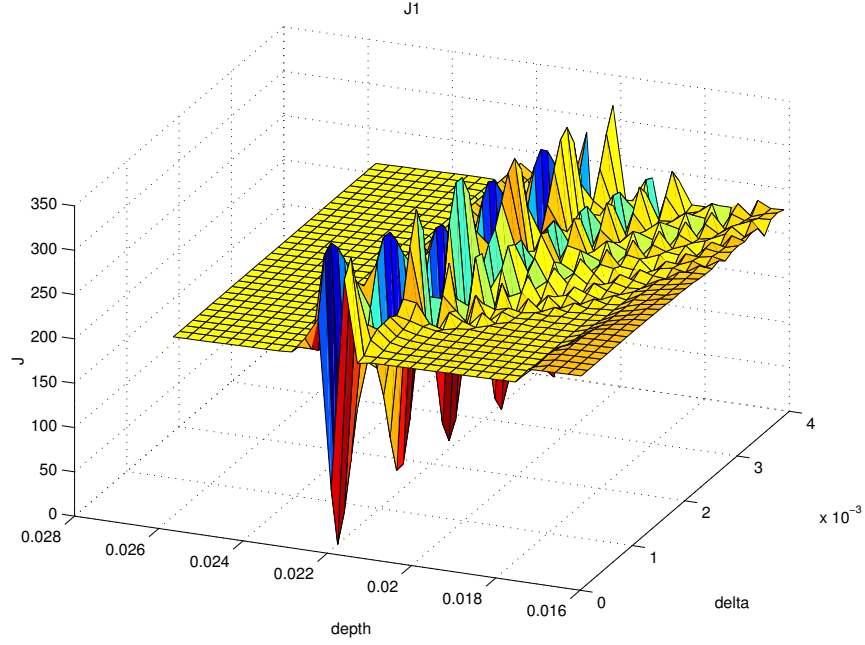


Figure 43: Close up surface plot of Least Squares objective function demonstrating peaks in J , and exhibiting many local minima.

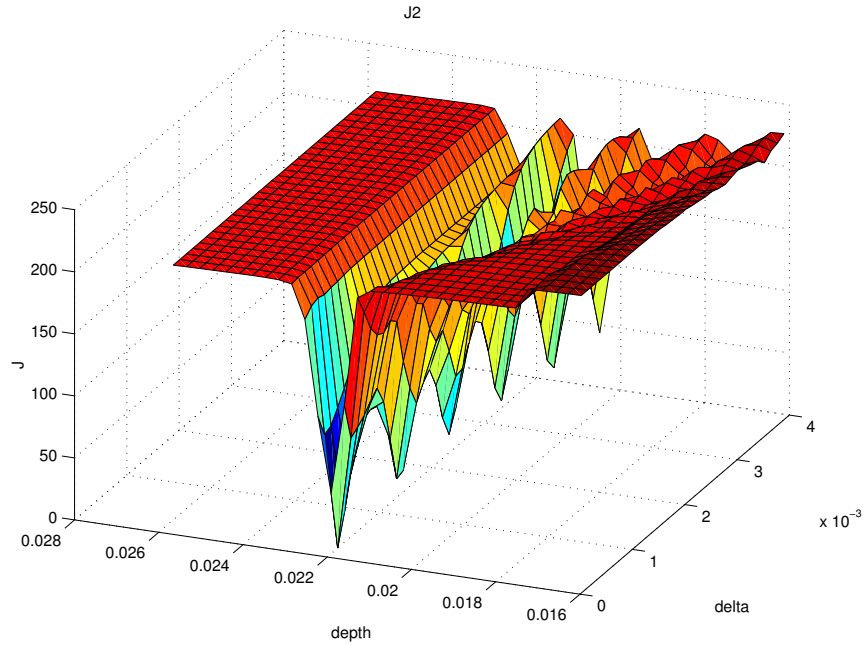


Figure 44: Close up surface plot of modified Least Squares objective function demonstrating lack of peaks in J , but exhibiting many local minima.

5.2 Initial Guesses

In spite of our faith in the “check point” method, we still desire to find the best initial guesses for our optimization routine as possible so that we may hopefully find the global minimum without restarting. As

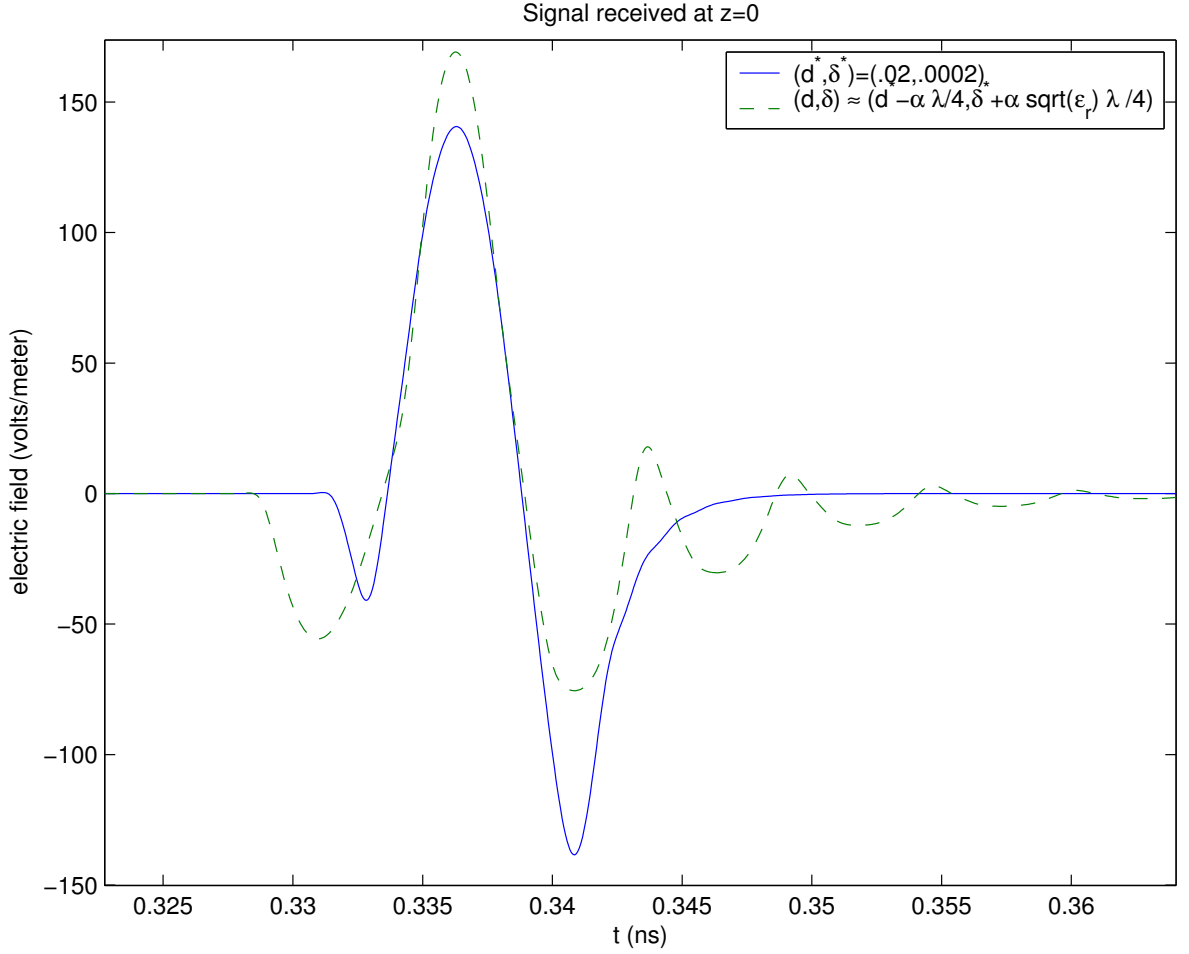


Figure 45: Data from (d^*, δ^*) and a simulation from the “check point” $(d^* - \alpha \frac{\lambda}{4}, \delta^* + \alpha \sqrt{\epsilon_r} \frac{\lambda}{4})$. The first trough cannot be matched, but δ is sufficiently large so that the signal’s peak matches with that of the data.

before, we use the travel time of the first trough to approximate the location of the first interface. However, in this formulation we can take advantage of some of the characteristics of the signals. For example, the first reflection off of the gap is always trough-first, and the second (as well as each subsequent reflection) is always peak-first. For this reason, if we want to locate the first trough we can simply find the largest peak (belonging to the second reflection) and back track. It is a very simple matter to find a maximum or minimum of a vector of values. After the location of the largest peak is found, we back track to find the minimum in front of it, namely that belonging to the first reflection off of the gap. Then using the procedure described in Section 4.1, we approximate the root immediately in front of this trough. That gives us the travel time for the first reflection off of the gap, which in turn gives us the depth d of the gap.

Finding δ is, unfortunately, not nearly as straightforward. There are two main possibilities, and therefore, two differing approaches to approximating δ :

- (i) The leading trough of the first reflection and the second reflection are disjoint (i.e., $\delta > \frac{\lambda}{8}$). In this case we can find the locations of the peak and trough and use the travel time between the two to approximate δ . We denote this approximation by δ_1 . (Note that the observed peak is not necessarily the same as the original peak unless $\delta > \frac{\lambda}{4}$, but it is still a good approximation). See Figure 47.
- (ii) The second reflection partially truncates the trough of the first (i.e., $\delta < \frac{\lambda}{8}$). As a rough approximation, we can assume that the location of the actual minimum (trough) is where the two signals begin to

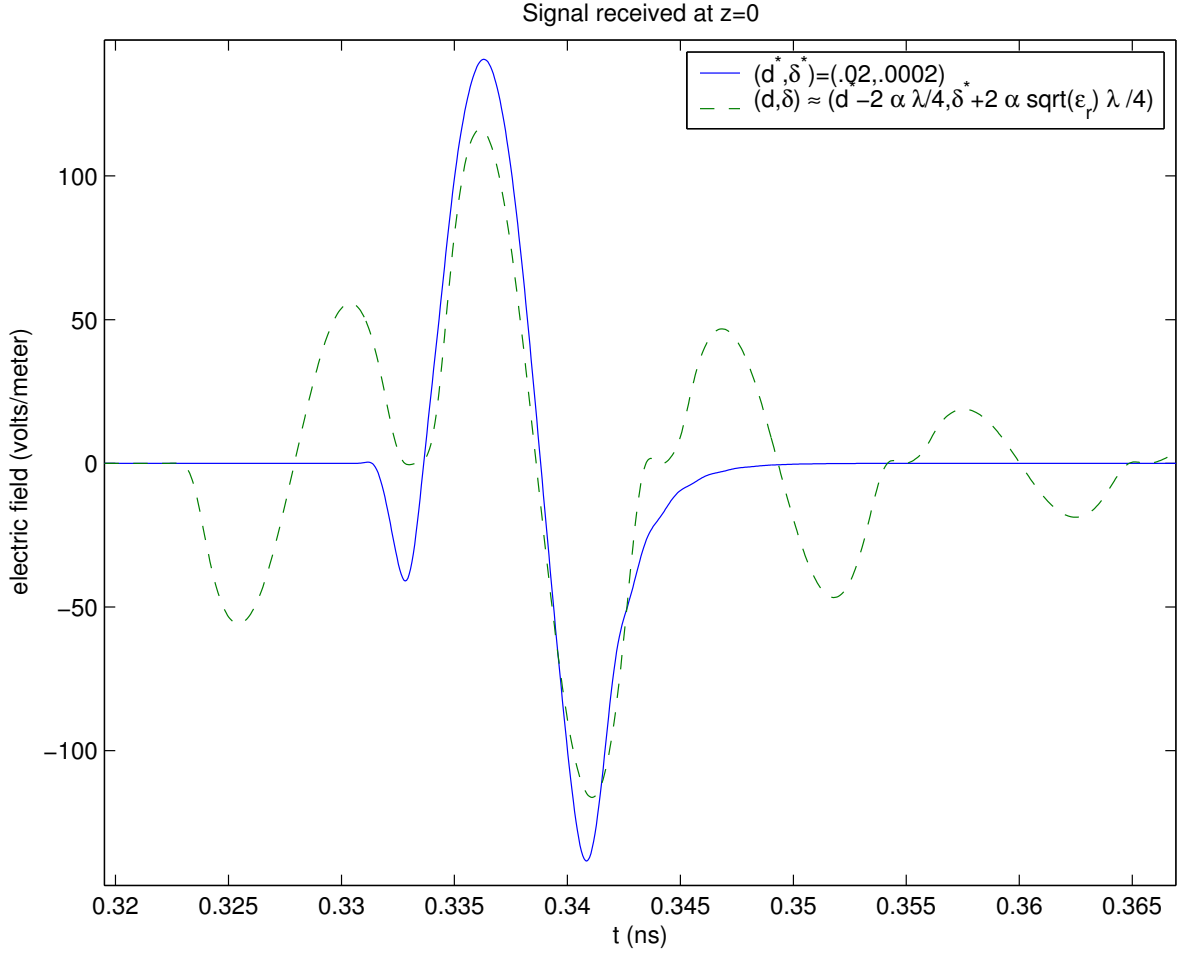


Figure 46: Data from (d^*, δ^*) and a simulation from the “check point” $(d^* - 2\alpha\frac{\lambda}{4}, \delta^* + 2\alpha\sqrt{\epsilon_r}\frac{\lambda}{4})$. Again, the first trough cannot be matched, but this time simulated signal has no cancellations so that its largest peak matches with that of the data.

interfere with each other (the observable minimum). See Figure 48. We denote this approximation by δ_4 .

A more accurate method is to use triangles to approximate the two reflections. By knowing the location of the maximum and minimum (peak and trough, respectively), and also the beginning of the first signal (from Section 4.1) and the rough approximation to the beginning of the second signal using δ_4 , we can estimate the slopes of the two triangles with finite differences. Also note that since the two signals are added, the observed root between the peak and trough in the combined signal is actually an equilibrium point between the two signals. By setting equal to each other the two linear approximations for each of the two signals, evaluated at the equilibrium point, we can solve for the distance between the starting point of each signal, and thus for δ_3 . See Figure 49. Specifically, let (p_1, q_1) be the location of the trough of the combined signal and (p_2, q_2) be the location of the peak. Let r_1 be the location of the root in front of the trough, and r_2 be the root between the trough and peak. Estimate the slope of the first signal, $m_1 < 0$, using (p_1, q_1) and r_1 . Now if we let $y = r_2 - r_1$ and say x is the actual distance between r_1 and the beginning of the second signal, then setting the linear approximations equal in magnitude, but opposite in sign, at r_2 yields

$$-m_1 y = m_2 (y - x).$$

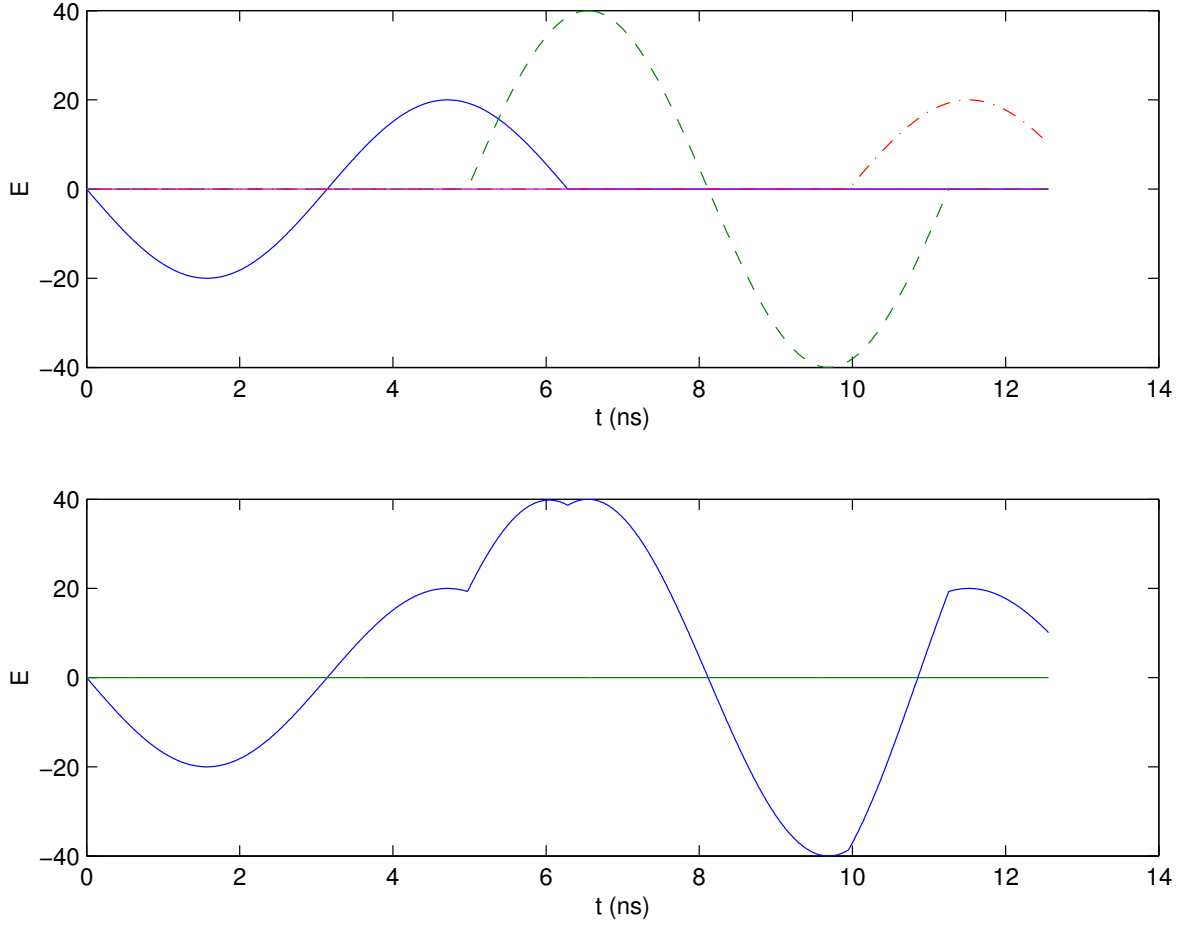


Figure 47: The top plot represents several signals which may be observed in a simulation of Problem 2. The bottom plot shows the sum of the top signals. The peak of the second signal is just beginning to be obscured by the first when δ becomes less than $\lambda/4$. Thus the observable maximum is still a good approximation of the peak of the second signal, and a trough to peak distance can be used to estimate δ .

Now we can estimate the slope of the second signal, $m_2 > 0$, using (p_2, q_2) and $(r_2, -m_1 y)$. Also, we can re-write the above equation as

$$x = \left(\frac{-m_1 + m_2}{m_2} \right) y.$$

To find δ_3 we simply divide x by 2 and the (scaled) speed of light in the material, i.e., $\sqrt{\epsilon_\infty}$.

Since each of the two situations above is dependent on the parameter it is approximating, we must also determine which of the above methods is most appropriate to use. Thus we use the most precise of the available methods to determine the situation, i.e., δ_4 , instead of δ_3 since in general δ_3 underestimates δ so we do not want to use it as a criterion for determining whether δ is small. (Note that when δ is indeed small, δ_3 is more accurate than δ_4 .) The estimate for δ_4 tends to be an overestimate, and is only valid if $\delta < \frac{\lambda}{8}$. Unfortunately, δ_1 also tends to be an overestimate, so we prefer to only trust it entirely if it is larger than $\frac{\lambda}{4}$. If neither δ_1 nor δ_3 is a sufficient approximation we choose to use the average of the two, and call it δ_2 .

Therefore our algorithm for approximating δ is as follows:

- (a) If $\delta_4 < \frac{\lambda}{8}$ then use δ_3
- (b) else if $\delta_1 > \frac{\lambda}{4}$ then use δ_1

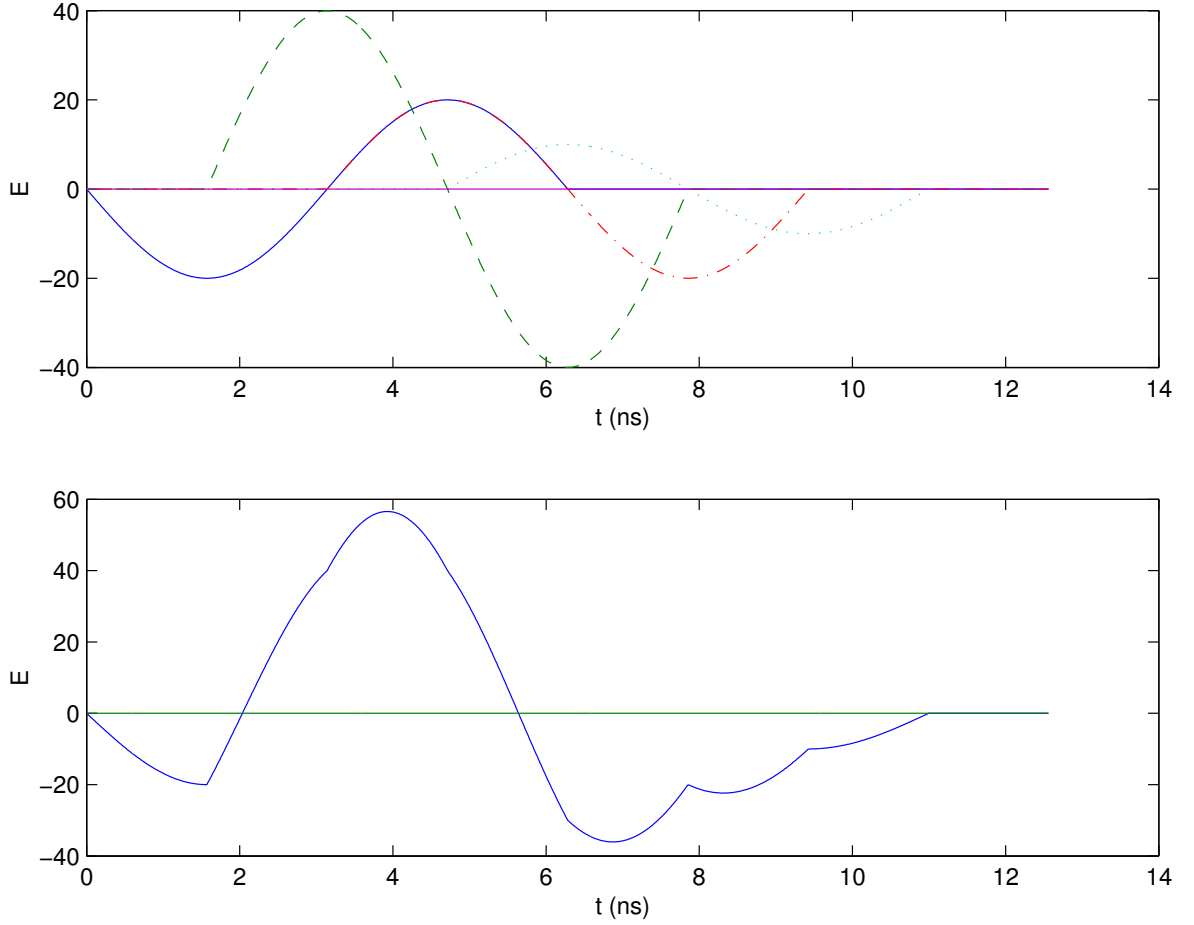


Figure 48: The top plot represents several signals which may be observed in a simulation of Problem 2. The bottom plot shows the sum of the top signals. The trough of the first signal is partially truncated by the second signal. In this case the observed minimum is still a good approximation to where the second signal begins. For smaller δ , a linear approximation must be used.

(c) else use δ_2 (average between δ_1 and δ_3).

We tested our approximating methods on exact depth (d) values of: .02, .04, .08, .1, and .2 m , and values of width (δ): .0001, .0002, .0004, .0006, and .0008 m . Since $\frac{\lambda}{8}$ is the transition point between the two situations, it is understandable why δ close to this value is the most difficult to accurately resolve. We chose this range of δ 's because our choice of frequency gives $\frac{\lambda}{8} = 3.7475 \times 10^{-4} m$. See Tables 2 and 3 for the initial estimates of d and δ respectively.

The approximations improve slightly as the number of finite elements is increased, and seemed to converge to fixed values. This suggests that numerical error (and instability) can affect the estimates. For each case there is a significant amount of numerical error in the simulations below a certain number of elements, therefore in approximating δ we chose to use the number of elements just above the threshold. The number of finite elements (N) used in the simulations is given for each case. Since approximating d does not depend on δ , we show the progression of estimates as N increases. The values in Table 2 which are bold denote the values used to approximate δ in Table 3.

These initial estimates may seem relatively inaccurate, some δ approximations being almost 100% off from the actual value, but in these tests all were sufficiently close to the global minima to not result in a false, local minima. While our “check point” method is available if needed, it is much more efficient to have an accurate initial estimate than to restart after optimizing from a bad one. Figures 50 through 54 display

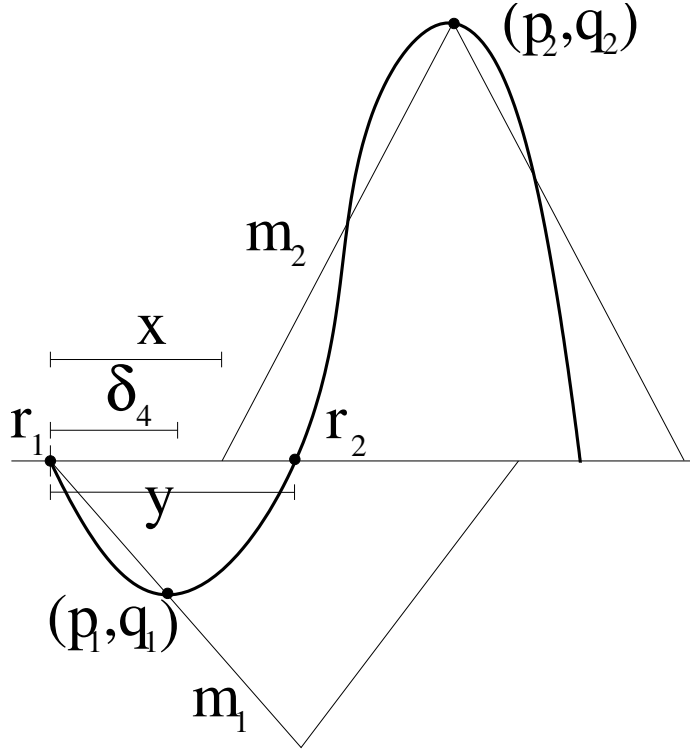


Figure 49: This schematic shows the roots, extrema, distances, and slopes used in the computation of δ_3 .

d	1024	2048	N 4096	8192	16384
.02	0.0200036	0.019996	0.0199998	0.0199998	
.04	0.0399689	0.0399766	0.0399919	0.0399958	
.08	0.0799225	0.0799609	0.0799839	0.0799992	
.1				0.0999799	
.2					0.200006

Table 2: The initial estimates of d . The values in bold denote the values used to approximate δ in Table 3.

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	9.41856e-05	0.000127645	0.000611592	0.000683544	0.000791472
.04	(N=2048)	0.000113806	0.000210178	0.000611592	0.000647568	0.000791472
.08	(N=4096)	2.89938e-05	0.00019437	0.000575616	0.000647568	0.000755496
.1	(N=8192)	0.000151182	0.000239409	0.000593604	0.000647568	0.000791472
.2	(N=16384)	8.74169e-05	0.000146323	0.000575616	0.000647568	0.000791472

Table 3: The initial estimates of δ .

surface plots of the modified Least Squares objective function zoomed in around the region that our initial estimates are located. It is evident that relatively small errors in the initial estimates could prevent an optimization routine from finding the global minima. For example, in Figure 50 if the initial estimate for d were 1×10^{-4} too high and δ too high by 2×10^{-4} then the gradient would cause both δ and d to increase, thus leading to the wrong minima. In cases such as these, the “check point” method provides a last resort.

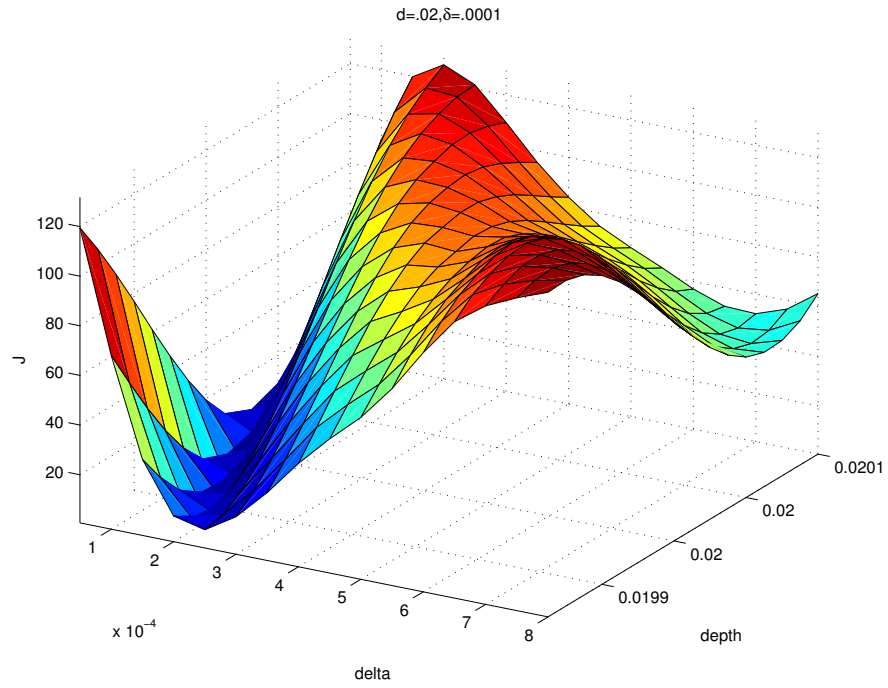


Figure 50: Very close surface plot of the modified Least Squares objective function using $d = .02$ and $\delta = .0001$.

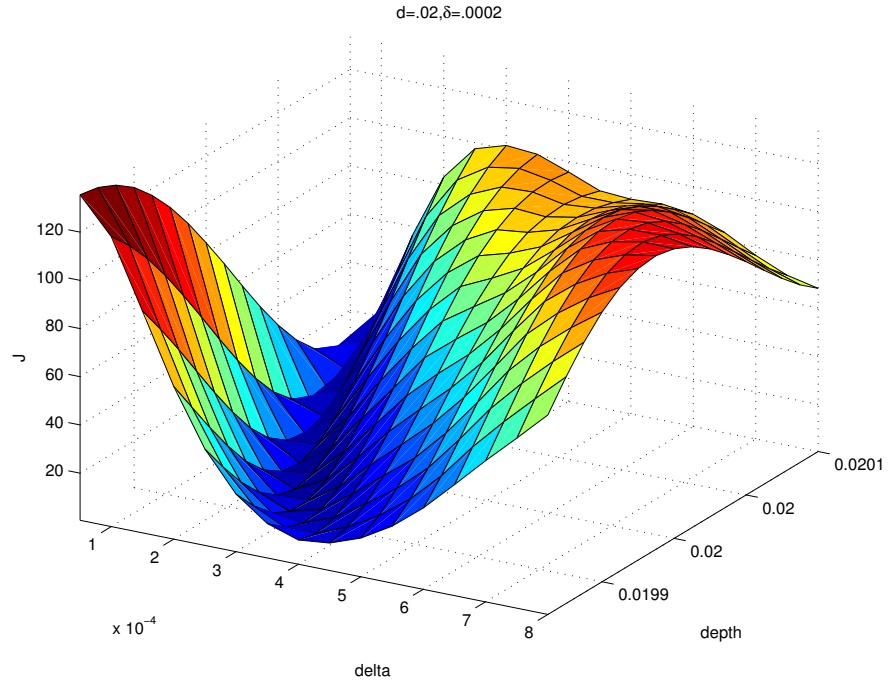


Figure 51: Very close surface plot of the modified Least Squares objective function using $d = .02$ and $\delta = .0002$.

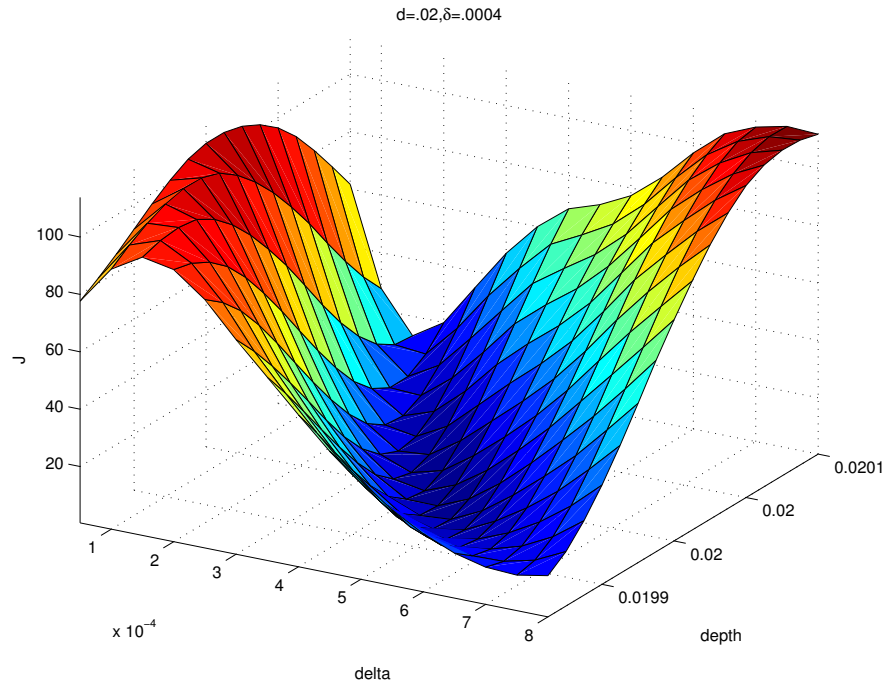


Figure 52: Very close surface plot of the modified Least Squares objective function using $d = .02$ and $\delta = .0004$.

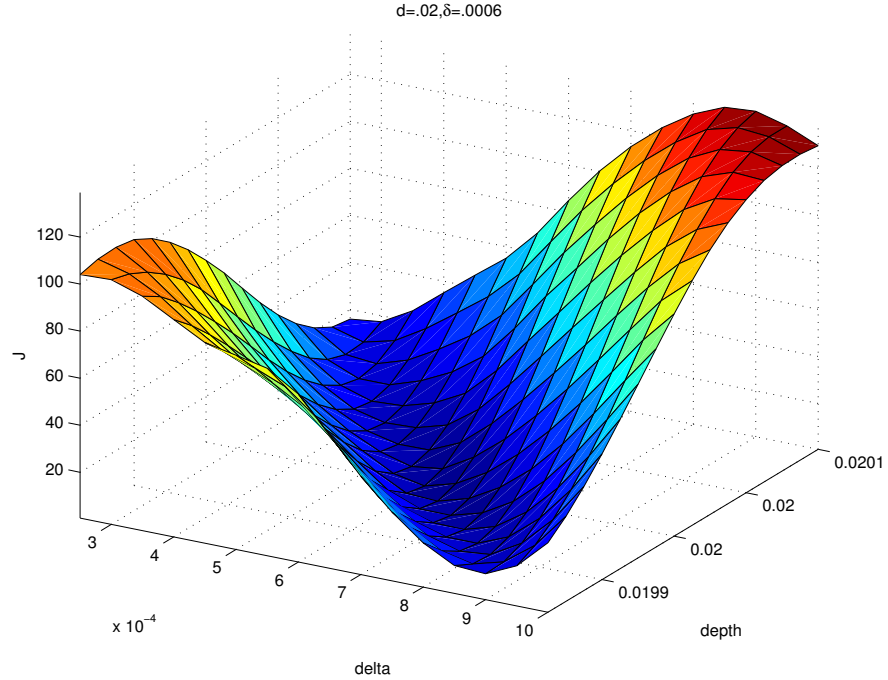


Figure 53: Very close surface plot of the modified Least Squares objective function using $d = .02$ and $\delta = .0006$. (Note that the axis is shifted from the previous Figures in order to include the minimum.)

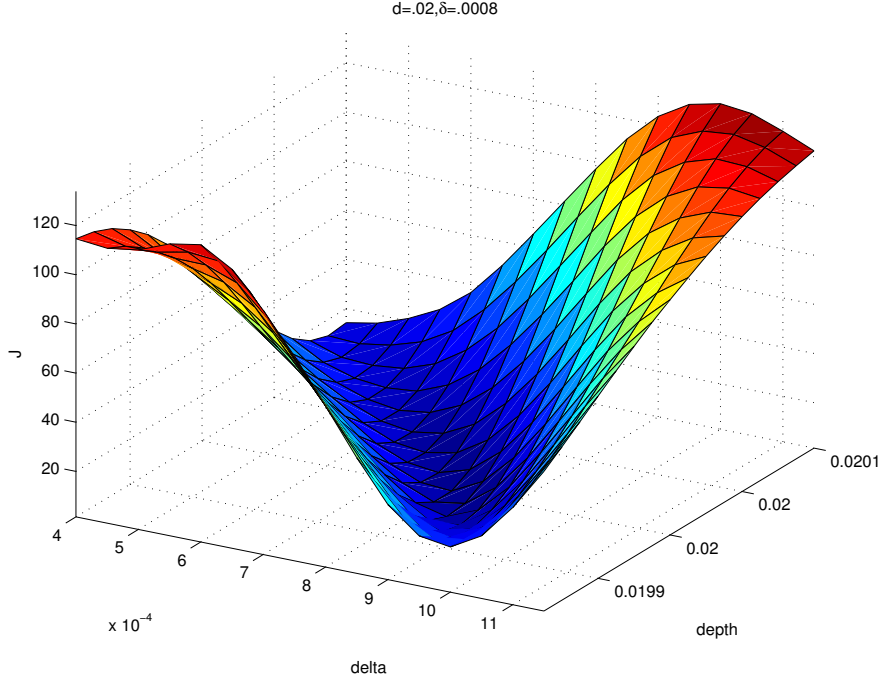


Figure 54: Very close surface plot of the modified Least Squares objective function using $d = .02$ and $\delta = .0008$. (Note that the axis is shifted from the previous Figures in order to include the minimum.)

5.3 Optimization Method

Now that we have approximated our initial guesses, we need to minimize the objective function in order to solve the inverse problem. In Problem 1, Gauss-Newton was sufficient to find the global minimum for most cases. In this formulation, however, we will apply more sophisticated methods, reverting to Gauss-Newton whenever possible since its convergence rate is best.

The first modification we make to Gauss-Newton is to add a Levenburg-Marquardt parameter, ν_c (see [7]). The Inexact Newton step becomes

$$s_c = - (R'(q_c)^T R'(q_c) + \nu_c I)^{-1} R'(q_c)^T R(q_c).$$

The parameter adds regularization by making the model Hessian positive definite. The method uses a quadratic model Hessian, and also has a built-in line search with a sufficient decrease condition. The line search is based on the predicted decrease computed from the quadratic model. If the actual improvement of the objective function, J , is close to the amount predicted by the model Hessian after a step is taken, then the method decreases the Levenburg-Marquardt parameter, ν_c , effectively increasing the relative size of the next step, which hopefully accelerates the convergence. As ν_c is decreased to 0 the method becomes Damped Gauss-Newton (meaning Gauss-Newton with a line search). If, however, the actual improvement of J after a step is not sufficient (or is even negative), ν_c is increased, effectively scaling back the Newton step, and we retest. If there are too many reductions then we declare a “line search failure” meaning that too small a step is required to decrease the objective function.

Usually a method would exit after a line search failure, returning the best approximation so far. But we use this failure to call an adaptive mesh size routine, i.e., an Implicit Filtering technique. The idea is that the failure is likely due to the fact that the direction the finite difference gradient chose is probably not an actual “descent direction” in the global sense. In other words, the finite differencing is most likely differentiating noise. In the same manner that a smooth surface may look rough under a microscope, using too small of a differencing step amplifies effects from round-off error and other sources of numerical noise. Our technique is to increase the relative differencing step, \hat{h} , recompute the gradients, and then try the

Levenburg-Marquardt method again. The relative differencing step, \hat{h} , is such that the gradient, $\nabla_{\hat{h}}$, of $J(q) = J([d, \delta])$ is computed with

$$\nabla_{\hat{h}} J([d, \delta]) = \begin{bmatrix} \frac{J((1+\hat{h})d, \delta) - J(d, \delta)}{\hat{h}d} \\ \frac{J(d, (1+\hat{h})\delta) - J(d, \delta)}{\hat{h}\delta} \end{bmatrix}$$

We apply a similar approach to modifying the differencing step \hat{h} as we do for changing ν_c in that after a successful step we decrease \hat{h} , but if we have another failure we increase \hat{h} even more. Since the convergence rates of gradient based methods are dependent on the size of \hat{h} (for example Gauss-Newton is $O(\hat{h}^2)$), we want \hat{h} to be as small as possible and still be effective, similarly with ν_c . We use a three tiered approach to changing \hat{h} . Initially we set $\hat{h} = 10^{-9}$. To increase \hat{h} we raise it to the $\frac{2}{3}$ power, to decrease we raise it to the $\frac{3}{2}$ power. Additionally we define 10^{-4} to be the maximum allowable differencing step value. Thus $\hat{h} \in \{10^{-9}, 10^{-6}, 10^{-4}\}$.

In general an optimization method exits with “success” if the norm of the current gradient is less than tol times the norm of the initial gradient. However, in our method we do not immediately trust the finite difference gradients, and instead call Implicit Filtering when the gradients appear small. When we have verified small gradients on all three scales (the various values of the differencing step \hat{h} defined above), then we exit with “success”.

Remark 4 *In practice, a very good solution is found within a couple of Levenberg-Marquardt steps, and then an equal number of Implicit Filtering iterations verify, and sometimes enhance, this solution. In the interest of efficiency, and since this is a parameter identification problem, we exit early with “success” if our objective function is satisfactorily small (i.e., tol times the initial value), thus saving at least half of the possible iterations.*

Additionally we impose a restriction on the number of “pullbacks” on each linesearch, and on the number of linesearches, effectively limiting the total number of iterations. If a small gradient has not been verified before exhausting the maximum number of iterations, we exit with “failure”.

5.4 Numerical Issues

For small N the difficult cases are those with large depth. This is because, as we mentioned before, the computational domain is effectively increased when the depth is increased, making the mesh sizes larger and increasing the level of numerical error. The magnitude of δ does not seem to have a significant effect on the convergence of the method.

An obvious disadvantage to having a large N is that each simulation takes much longer. Although on average fewer function calls are required with the larger N cases to get the same level of accuracy, in general the total execution time is quadrupled when the number of elements is doubled. This is consistent with the fact that complexity of the most time consuming part of the simulation, the linear solves, is $O(N)$, and the number of time steps N_t is also $O(N)$. So when we double the number of finite elements we are also doubling the number of time steps. Therefore, we get an overall complexity of $O(N^2)$. Thus, as mentioned before, in our inverse problem we choose to use the number of elements just above the threshold of when numerical error is apparent.

We should also mention that in order to create data, in lieu of actual experimental data, we perform a simulation at a higher resolution believing it to be more accurate. Specifically, we double the number of finite elements. Since the time step, and therefore the effective sample rate if the time step is too large, are both dependent upon the mesh size (recall $\Delta t = O(h)$ and Remark 2), the sample times of the simulated data do not necessarily correspond with the sample times of the simulations at the lower resolution. (In general we have twice as many samples from the higher resolution.) Thus in order to compute the modified least squares error between the two vectors, we perform a linear interpolation of the simulated data onto the sample times at the lower resolution. See Figure 55. Note that in the usual case where we simply have twice as many sample points from the higher resolution simulation, we are in effect discarding sample points rather than doing a true interpolation.

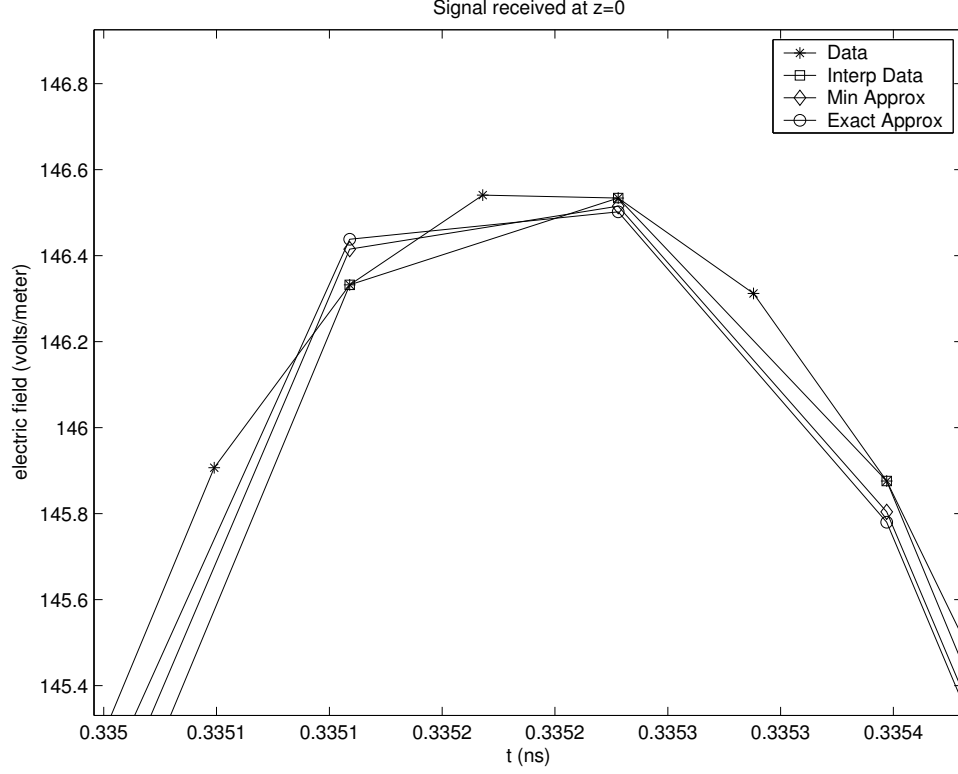


Figure 55: Plotted are the actual simulated data ($N = 2048$), the interpolation of the simulated data onto the low resolution sample times ($N = 1024$), the result of the minimization routine ($N = 1024$), and a low resolution ($N = 1024$) simulation using the exact values of d and δ .

For comparison we compute the low resolution simulation using the values d^* and δ^* (note that this is not the same as taking the high resolution simulation and interpolating it onto the low resolution time steps, which we actually use as our observed data). In every case that we have tested, J , when computed with the d and δ values found from the optimization routine (d_{min} and δ_{min}), is less than or equal to J when computed with the original values (d^* and δ^*). This suggests that an actual global minimum of the objective function has been found, even though the final estimates of d and δ themselves are not necessarily equal to d^* and δ^* . Note in Figure 55 that the simulation using original values, (d^*, δ^*), is in fact closer to the original data, but the simulation using the minimizer values, (d_{min}, δ_{min}), is closer to the interpolated data (see for example the $[.335, .3352]$ interval).

Although we could compute our optimization routine at the same resolution as the simulated data to get a better fit in our tests, this would not properly represent the real-life phenomenon of sampling data. Sampled data is inherently not a completely accurate representation of a physical observation. We believe that our interpolation approach gives a more realistic expectation of how our method would perform given actual experimental data. In order to further test the robustness of our inverse problem solution method we introduce random noise to the detected data in Section 5.5.

5.5 Numerical Results

Tables 4 and 5 show the final computed approximations for the depth of the slab (d_{min}) and the width of the gap behind it (δ_{min}). The relative differences from the original values used to generate the data (d^* and δ^*), are: for depth, on the order of .0001 and for δ , on the order of .01. However, this does not imply that the optimization routine was unable to find the optimal solution. Recall that since our data is generated

with essentially a different simulator than our forward solves, the original values do not necessarily minimize the objective function. The objective function values give a better indicator of how well the optimization routine works since it shows the fit to the generated data. Table 6 shows the final objective function values. In each of these cases, the final objective function value (J_{min}) was less than $J^* := J(q^*)$. In fact, the ratios $J_r := J_{min}/J^*$ were on average .3008. We consider any $J_r < 1$ to represent a successful convergence.

Although δ values that are near $\frac{\lambda}{8} = 3.7475 \times 10^{-4}m$ are the most difficult for which to obtain initial approximations, we see that the objective function values in these cases are just as small (and the final estimates are just as close) as for other δ values.

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	0.0200053	0.0200022	0.0200006	0.0200005	0.0200002
.04	(N=2048)	0.0399948	0.0399974	0.0400005	0.0400005	0.0399999
.08	(N=4096)	0.0799973	0.0799987	0.0800006	0.0800006	0.0800003
.1	(N=8192)	0.0999945	0.0999974	0.1	0.1	0.0999999
.2	(N=16384)	0.200011	0.200005	0.2	0.2	0.200001

Table 4: The final estimates of d .

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	9.40622e-05	0.000196754	0.000398642	0.000597275	0.00079707
.04	(N=2048)	0.000106435	0.000203916	0.000394204	0.000592156	0.000793622
.08	(N=4096)	0.000103585	0.000202273	0.000395791	0.000593861	0.000794401
.1	(N=8192)	0.000106593	0.000203876	0.000396203	0.000594976	0.000795985
.2	(N=16384)	8.7456e-05	0.000191808	0.00040297	0.000602902	0.00080129

Table 5: The final estimates of δ .

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	0.00786171	0.00906699	0.0115657	0.0233783	0.0447687
.04	(N=2048)	0.021516	0.0343314	0.0514108	0.0700747	0.0927117
.08	(N=4096)	0.0116105	0.0145428	0.0201004	0.0272513	0.0344458
.1	(N=8192)	0.00304723	0.00547532	0.00779186	0.00931778	0.0118529
.2	(N=16384)	0.000609258	0.00133978	0.00146975	0.000962975	0.000766141

Table 6: The objective function value of the final estimates.

The execution time, in seconds, is given in Table 7. While the previous tables show that we were actually able to resolve the case of 20cm depth, we see here what price we had to pay. The average execution times for each of different mesh sizes ($N = 1024, 2048, 4096, 8192$, and 16384) were 39, 248, 1452, 6229, and 35509 seconds, respectively. Each represents an increase in time over the previous mesh size by a factor of 6.4, 5.9, 4.28, and 5.7, respectively. This is consistent with the fact that the forward solves are order $O(h^2)$. However, the additional sample points for the larger N cases allowed for smaller objective function values which took increasingly more iterations to satisfy the relative tolerance in our stopping criteria. This explains why we do not see ratios closer to the expected 4 for order $O(h^2)$ methods. The number of calls to the simulator (related to the number of Levenburg-Marquardt iterations) for each case is given in Table 8.

5.5.1 Relative Random Noise

We add random noise to the signal, as mentioned above, in order to more closely simulate the experimental process in data collection. As in Section 4.3.2, we start with relative noise where the absolute value of the noise is proportional to the size of the signal. If E_i is the data sampled, then we define $\hat{E}_i = E_i(1 + \nu_r \eta_i)$,

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	39	35	48	38	35
.04	(N=2048)	198	197	319	196	328
.08	(N=4096)	1560	1458	1500	1071	1671
.1	(N=8192)	6141	5256	7625	5319	6802
.2	(N=16384)	31818	42755	35860	26521	40593

Table 7: The execution time in seconds to find the final estimates.

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	23	23	33	26	23
.04	(N=2048)	24	24	43	24	45
.08	(N=4096)	38	35	36	24	41
.1	(N=8192)	29	24	37	24	33
.2	(N=16384)	36	50	41	29	48

Table 8: The total number of calls to the simulator required to find the final estimates.

where η_i are independent normally distributed random variables with mean zero and variance one. Again, the coefficient ν_r determines the relative magnitude of the noise as a percentage of the magnitude of E_i , in particular, $\nu_r = 0.01$ corresponds to 2% noise. We tested relative magnitude levels of 2%, 10%, and 20% (corresponding to $\nu_r = .01, .05$, and $.1$ respectively). Table 9 shows the initial estimates of d for $\nu_r = .1$ (other ν_r values produced very similar results, so we only give the worst case here). Tables 10 through 12 show the initial estimates of δ for several different noise levels. There is not a noticeable significant change in the accuracy of the estimates even for $\nu_r = .1$. In nearly all the cases the estimate was close enough for the optimization method to converge ($J_r < 1$) to the expected minimum. The only exceptions were with $\nu = .1$ and $\delta = .0004$, which are understandably the most difficult cases.

The final approximations d_{min} and δ_{min} in the presence of noise are given in Tables 13 through 18. Some approximations with high noise may appear to be better approximations than some with little or no noise. For example, with $\delta^* = .0001, d^* = .04$, the $\nu_r = .1$ approximations are an order of magnitude closer to the original values than the $\nu_r = 0$ approximations. This is not to say that the noise helps the approximation method. Rather, it is for the same reason that, for example, as shown in Figure 55, the actual parameter values produced a signal farther away (in the Least Squares sense) from the generated data than a signal computed with the approximated parameter values. The resulting objective function values give a better indication of the accuracy of the approximation to the data. A comparison of Tables 6 and 19 clearly show that the data without noise is more accurately matched by its approximations than those with noise.

d	N		
	1024	2048	4096
.02	0.020019	0.0199883	0.0199998
.04	0.0399689	0.0399766	0.0399919
.08	0.0799225	0.0800069	0.0799992

Table 9: The initial estimates of d with $\nu_r = .1$. The values in bold denote the values used to approximate δ in Table 12. (Initial estimates of d using other ν_r values were very simliar and therefore are omitted.)

d	δ				
	.0001	.0002	.0004	.0006	.0008
.02 (N=1024)	9.43492e-05	0.000127736	0.000611592	0.000647568	0.000791472
.04 (N=2048)	0.000115174	0.000209026	0.000611592	0.000647568	0.000755496
.08 (N=4096)	3.20852e-05	0.000194313	0.000611592	0.000647568	0.000827448

Table 10: The initial estimates of δ with $\nu_r = .01$.

d	δ				
	.0001	.0002	.0004	.0006	.0008
.02 (N=1024)	4.34312e-05	0.000129248	0.000575616	0.000575616	0.000755496
.04 (N=2048)	0.000115175	0.000211701	0.000611592	0.000755496	0.000755496
.08 (N=4096)	1.76356e-05	0.000194415	0.000647568	0.000647568	0.000683544

Table 11: The initial estimates of δ with $\nu_r = .05$.

d	δ				
	.0001	.0002	.0004	.0006	.0008
.02 (N=1024)	0.000107928	0.000172127	0.000575616	0.000323784	0.000611592
.04 (N=2048)	9.99018e-05	0.000216157	0.000575616	0.000611592	0.000647568
.08 (N=4096)	4.76434e-05	0.000194762	0.000611592	0.000647568	0.000755496

Table 12: The initial estimates of δ with $\nu_r = .1$.

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	0.0200057	0.0200021	0.0200006	0.0200008	0.0200001
.04	(N=2048)	0.0399944	0.0399971	0.0400005	0.0400005	0.0399998
.08	(N=4096)	0.0799997	0.0799986	0.0800004	0.0800005	0.0800003

Table 13: The final estimates of d using $\nu_r = .01$.

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	0.0200061	0.0200009	0.0200016	0.0200013	0.0200002
.04	(N=2048)	0.0399951	0.0399982	0.0399993	0.0400006	0.0399997
.08	(N=4096)	0.0799957	0.0799971	0.0799999	0.0800004	0.0800001

Table 14: The final estimates of d using $\nu_r = .05$.

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	0.0200059	0.020006	0.0200017	0.0199998	0.0199974
.04	(N=2048)	0.0399993	0.0399976	0.0399991	0.0400021	0.0399982
.08	(N=4096)	0.0799982	0.079997	0.0800002	0.0799986	0.0800013

Table 15: The final estimates of d using $\nu_r = .1$.

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	9.34307e-05	0.000196985	0.000398454	0.000596308	0.000797028
.04	(N=2048)	0.000106823	0.000204701	0.000394249	0.0005922	0.000793984
.08	(N=4096)	0.000103996	0.000202579	0.000396222	0.000593539	0.000794799

Table 16: The final estimates of δ using $\nu_r = .01$.

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	9.40065e-05	0.000198868	0.000396695	0.000597252	0.000794729
.04	(N=2048)	0.000105547	0.000203208	0.000400938	0.000592641	0.000791008
.08	(N=4096)	0.000105656	0.000203992	0.000396485	0.000594405	0.000793268

Table 17: The final estimates of δ using $\nu_r = .05$.

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	9.34486e-05	0.000188052	0.000392001	0.000594642	0.00079958
.04	(N=2048)	0.000100242	0.000204073	0.000401208	0.000586155	0.000798636
.08	(N=4096)	0.000101771	0.000205295	0.000400441	0.000604353	0.000798047

Table 18: The final estimates of δ using $\nu_r = .1$.

d		δ				
		.0001	.0002	.0004	.0006	.0008
.02	(N=1024)	3.89043	4.0585	1.64814	3.46279	4.1071
.04	(N=2048)	1.29514	1.42229	1.47275	1.5261	1.61906
.08	(N=4096)	0.779184	0.717672	0.848017	0.647817	0.674641

Table 19: The objective function value of the final estimates using $\nu_r = .1$.

5.5.2 Standard Error Analysis

In an actual inverse problem using data collected by experiment, one needs to have confidence intervals on all parameter estimates. We will apply standard error techniques to an Ordinary Least Squares (OLS) formulation of our problem to obtain confidence intervals on our estimates. In order to rewrite our objective function in an OLS formulation, we define $y(t; q) = |E(t, 0; q)|$ to be our estimate to $\hat{y} = |\hat{E}|$, which is the data we are trying to fit by determining $q = (d, \delta)$. Now it is clear that our objective function can be written in the standard OLS form

$$J(q) = \frac{1}{N_s} \sum_{i=1}^{N_s} |y(t_i; q) - \hat{y}_i|^2.$$

For simplicity of terminology, in this section alone, we will refer to $|\hat{E}_i|$ as the data and to $|E(t_i, 0; q)|$ as the simulations.

With the relative random noise described above we do not have constant variance, as is demonstrated in Figures 56 and 57. Here we have plotted the residual $r_i := |E(t_i, 0; \hat{q}_{OLS})| - |\hat{E}_i|$ against time, t_i , and also against $|E(t_i, 0; \hat{q}_{OLS})|$. As one would expect with noise that is relative in size to the signal value, we have a pattern in Figure 56 that follows the pattern of the original signal. Figure 57 demonstrates the fan shape associated with noise that is dependent upon the size of the signal, i.e., non constant variance.

Since constant variance is most conveniently assumed in standard error analysis, we further consider estimates obtained from an inverse problem applied to data with constant variance random noise added. In particular, the data we now consider is generated by

$$\hat{E}_j = E(t_j, 0; q^*) + \beta \nu_r \eta_j$$

where

$$\eta_j \sim \mathcal{N}(0, 1)$$

and the constant β is a scaling factor chosen simply so that the noise level, ν_r , will somewhat correspond to the parameter ν_r used in the previous section on relative noise. Specifically, $\beta = \max_i \hat{E}_i / 10$ ensures that J^* in the constant variance cases is on the same order of magnitude as those in the relative noise cases above for all choices of d and δ that we have considered.

The variance of this data is

$$\sigma^2 = \mathcal{E}[\beta^2 \nu_r^2 \eta_j^2] = \beta^2 \nu_r^2 \mathcal{E}[\eta_j^2] = \beta^2 \nu_r^2$$

where \mathcal{E} denotes the expectation. Therefore, we do have constant variance. Note further the resulting lack of patterns in Figures 58 and 59. The suspicious looking phenomenon of many points on the line $E = 0$ is simply because in the original data E is very close to zero most of the time. (Recall Figures 45 and 46 for example.) Figure 60 demonstrates graphically the difference between relative noise and constant variance noise. The relative noise case is particularly difficult in our inverse problem since most of our initial estimates are based on accurately determining the peak locations, yet this is exactly where most of the relative noise is concentrated.

With constant variance, and further, assuming that each η_j is identically independently (normally) distributed, we have that (see [4]) in the limit as $N_s \rightarrow \infty$

$$\hat{q}_{OLS} \sim \mathcal{N}_2 \left(q_0, \sigma_0^2 [\mathcal{E}^T(q_0) \mathcal{E}(q_0)]^{-1} \right).$$

Here $\mathcal{E}(\hat{q}) = \frac{\partial |E|}{\partial q}(\hat{q})$ which is an $N_s \times 2$ matrix since $q = (d, \delta)$ and $|E|$ is evaluated at N_s sample times. Also, the scale parameter σ_0^2 is approximately given by

$$\sigma_0^2 = \frac{1}{N_s - 2} \sum_{i=1}^{N_s} \left(|E(t_i, 0; q_0)| - |\hat{E}_i| \right)^2.$$

In the above equations, q_0 denotes the theoretical “true” value of the parameter that best describes the system from which the data is taken. Note that in this case, this is not necessarily the same as q^* since the

method used to generate the data is different from the forward solve simulator. Therefore q_0 is generally unknown even in examples with simulated data.

As demonstrated in the previous sections, our \hat{q}_{OLS} is often a better minimizer than even the original value of q^* , therefore we will approximate q_0 in the above equations by \hat{q}_{OLS} . In particular, if we denote the covariance matrix as $C_0 = \sigma_0^2 [\mathcal{E}^T(q_0)\mathcal{E}(q_0)]^{-1}$, then we will approximate C_0 by $C = \sigma_{OLS}^2 [\mathcal{E}^T(\hat{q}_{OLS})\mathcal{E}(\hat{q}_{OLS})]^{-1}$, where

$$\sigma_{OLS}^2 = \frac{1}{N_s - 2} \sum_{i=1}^{N_s} \left(|E(t_i, 0; \hat{q}_{OLS})| - |\hat{E}_i| \right)^2.$$

We compute σ_{OLS}^2 by multiplying our J_{min} by an appropriate conversion factor, since they are defined in a similar manner. However, in order to compute the partial derivatives with respect to d and δ in \mathcal{E} we employ forward differencing, which requires an additional forward simulation for each q_j . For $\hat{q} = \hat{q}_{OLS}$ we have, for example

$$\mathcal{E}_{i1} = \frac{\partial |E|}{\partial q_1}(t_i, 0; \hat{q}) \approx \frac{|E(t_i, 0; [\hat{q}_1, \hat{q}_2])| - |E(t_i, 0; [(1 - h_d)\hat{q}_1, \hat{q}_2])|}{h_d \hat{q}_1}$$

and similarly for each \mathcal{E}_{i2} . In our computations we used the relative differencing factor of $h_d = 1 \times 10^{-4}$. One could also use a sensitivity equations approach (e.g., see [1] and the references therein), but since the variational equations are quite difficult to solve for this example, we choose instead to approximate the partials with respect to q directly with our simulations.

We also need to point out that while taking the absolute value of a function limits differentiability at a small number of points, the derivative does exist almost everywhere. The absolute value function does not change the magnitude of the derivative where it exists, which is what we need to compute the dot product of \mathcal{E} with itself. By using finite differences to estimate derivatives, we are essentially under-estimating at the discontinuities. Under-estimating a few points out of thousands is not going to significantly change our covariance matrix. (Alternatively, one could have defined the objective function by squaring the signals instead of taking absolute values to avoid this problem. In this research we were interested in comparing J_1 and J_2 in previous sections above and changing the scale of E by squaring it would have prevented this.)

With \mathcal{E} calculated, we can now evaluate $C = \sigma_{OLS}^2 [\mathcal{E}^T(\hat{q}_{OLS})\mathcal{E}(\hat{q}_{OLS})]^{-1}$. Then the standard error for $q_1 = d$ is estimated by $\sqrt{C_{11}}$ while the standard error for $q_2 = \delta$ is estimated by $\sqrt{C_{22}}$. See Tables 20 through 27 for confidence intervals relating to various d^* , δ^* and ν_r values. For example, in the case of $d^* = .02$, $\delta^* = .0002$ and with $\nu_r = .01$ our covariance matrix is

$$C = \begin{bmatrix} 2.37122 \times 10^{-15} & -4.43815 \times 10^{-15} \\ -4.43815 \times 10^{-15} & 9.1829 \times 10^{-15} \end{bmatrix}$$

which results in the confidence intervals $d \in (2.00004 \pm 4.86952 \times 10^{-6}) \times 10^{-2}$ and $\delta \in (1.9941 \pm 0.000958274) \times 10^{-4}$.

The width of these bounds are $\pm 0.000243471\%$ and $\pm 0.0480555\%$ of the approximation value respectively. For the $d^* = .02$ case, the average size of the confidence intervals for $\nu_r = .01, .05, .1$ respectively were $\pm .0002\%, \pm .001\%, \pm .002\%$ (averaged over various δ^* values ranging from .0001 to .0008). It is interesting that the widths of the confidence intervals nearly exactly double, on average, when the noise level doubles. For the $d^* = .04$ case the average size of the confidence intervals were $\pm .0001\%, \pm .0006\%, \pm .001\%$. Likewise, when the widths of the confidence intervals for $\delta^* = .0002$ are averaged over several various d^* values (.02, .04, .08) we get $\pm .05999\%, \pm .2883\%, \pm .5718\%$ for $\nu_r = .01, .05, .1$ respectively. For $\delta^* = .0004$ the averages are $\pm .03331\%, \pm .1575\%, \pm .3154\%$. In general, larger d^* and δ^* values have smaller (tighter) confidence intervals. This suggests that the approximations found in these cases are better than those estimating small parameters. While this is intuitive, it is not apparent looking at the estimates themselves or even the final objective function values (see, for example, Table 6).

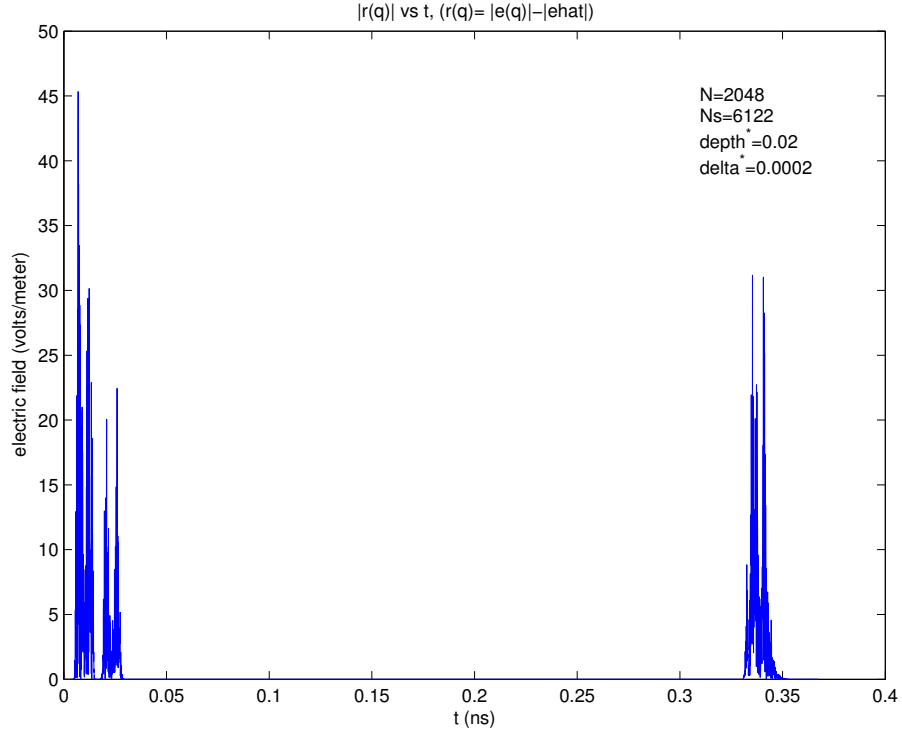


Figure 56: Plots of the absolute value of the residual $r_i = |E(t_i, 0; \hat{q}_{OLS})| - |\hat{E}_i|$ versus time t_i when the data contains relative random noise.

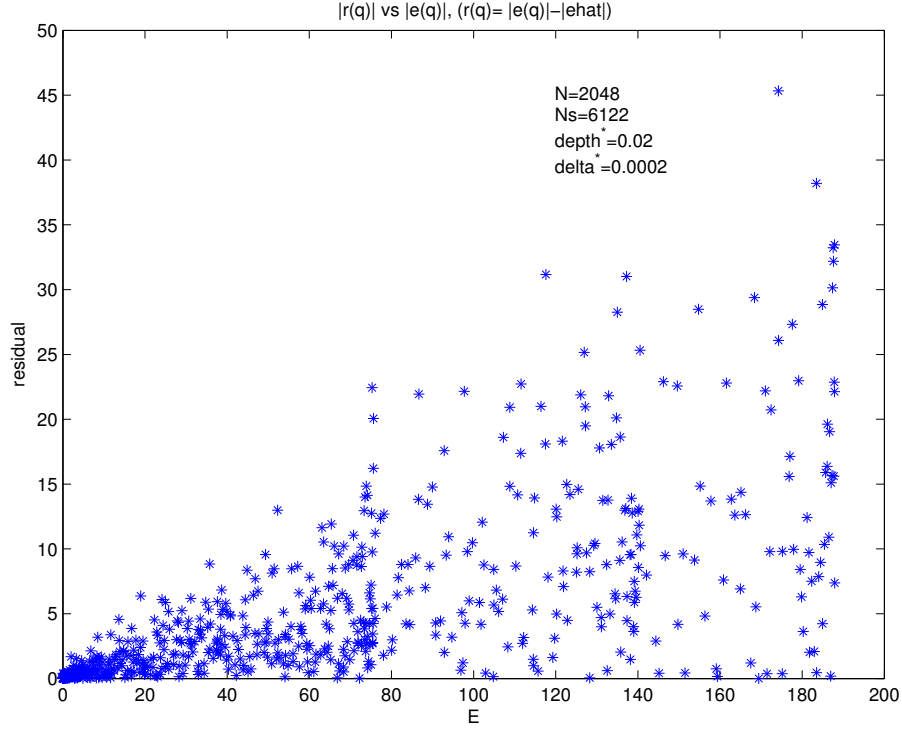


Figure 57: Plots of the absolute value of the residual $r_i = |E(t_i, 0; \hat{q}_{OLS})| - |\hat{E}_i|$ versus the absolute value of the electric field $E(t_i, 0; \hat{q}_{OLS})$ when the data contains relative random noise.

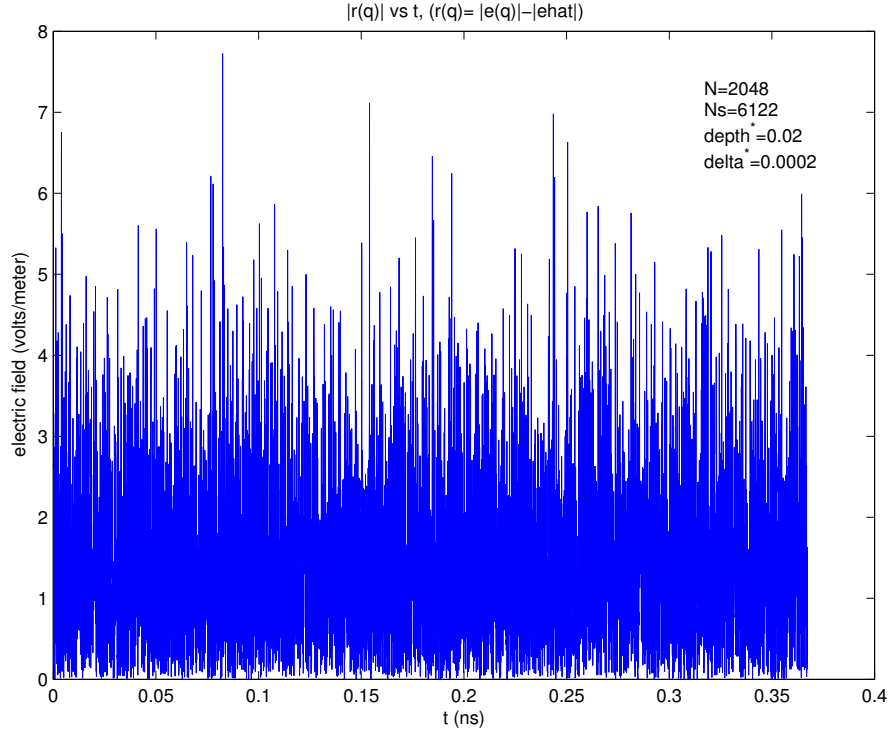


Figure 58: Plots of the absolute value of the residual $r_i = |E(t_i, 0; \hat{q}_{OLS})| - |\hat{E}_i|$ versus time t_i when the data contains constant variance random noise.

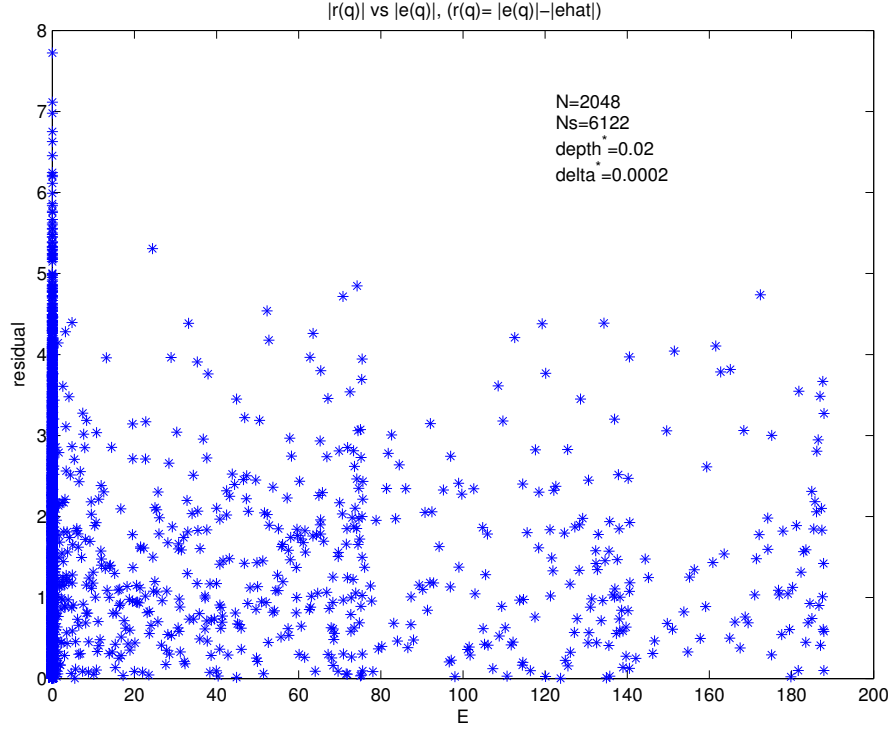


Figure 59: Plots of the absolute value of the residual $r_i = |E(t_i, 0; \hat{q}_{OLS})| - |\hat{E}_i|$ versus the absolute value of the electric field $E(t_i, 0; \hat{q}_{OLS})$ when the data contains constant variance random noise.

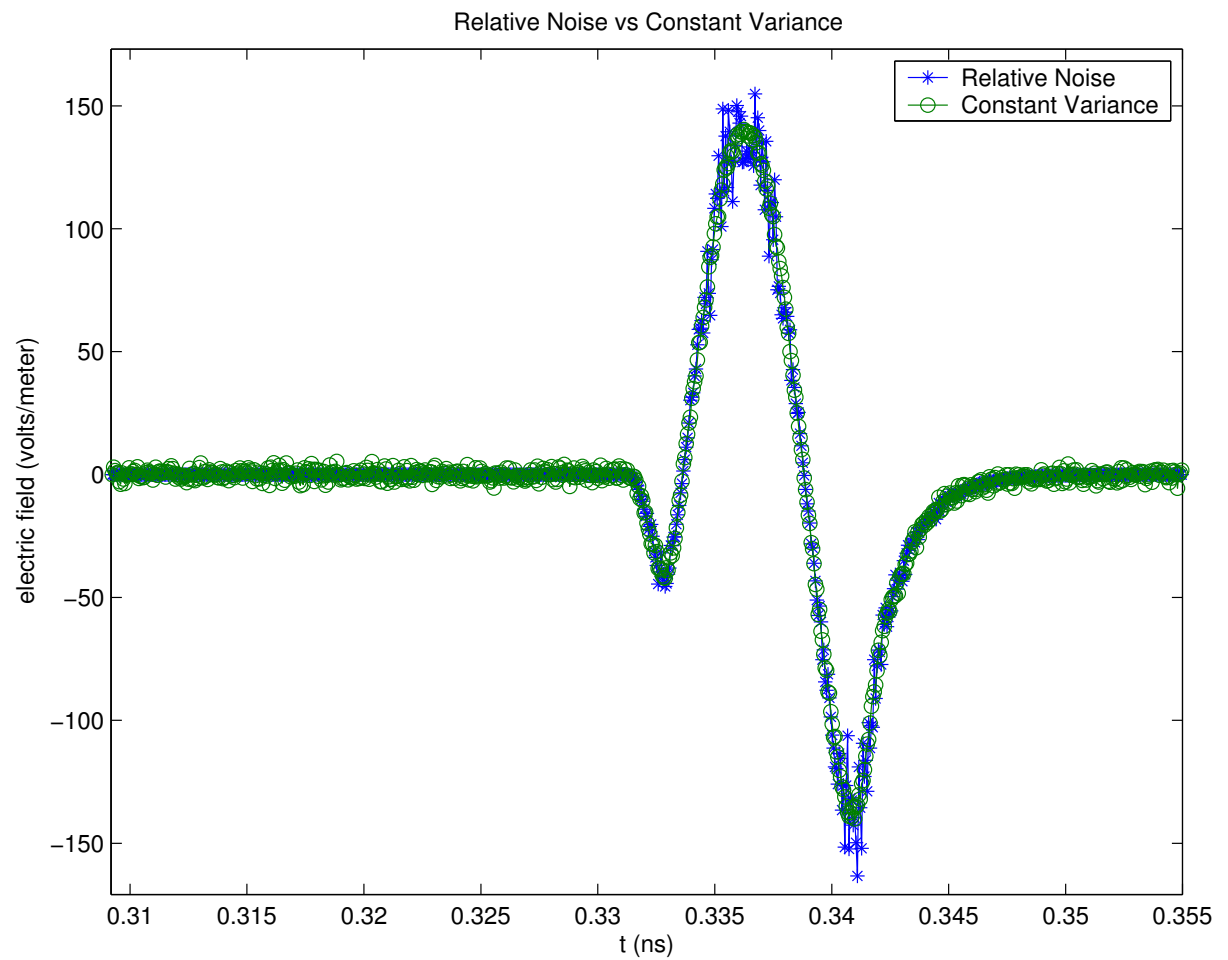


Figure 60: The difference between data with relative noise added and data with constant variance noise added is clearly evident when E is close to zero or very large.

δ	$d^* = .02$ ($N = 2048$)	$d^* = .04$ ($N = 4096$)
.0002	$(2.00005 \pm 9.30284 \times 10^{-7}) \times 10^{-2}$	$(4.00013 \pm 1.62162 \times 10^{-6}) \times 10^{-2}$
.0004	$(2.00001 \pm 6.50411 \times 10^{-7}) \times 10^{-2}$	$(4.00001 \pm 1.19064 \times 10^{-6}) \times 10^{-2}$
.0008	$(2.00001 \pm 4.91232 \times 10^{-7}) \times 10^{-2}$	$(4.00002 \pm 9.05240 \times 10^{-7}) \times 10^{-2}$

Table 20: Confidence intervals for the OLS estimate of d when the data is generated with no noise (i.e., $\nu_r = 0.0$).

δ	$d^* = .02$ ($N = 2048$)	$d^* = .04$ ($N = 4096$)
.0002	$(2.00004 \pm 4.86952 \times 10^{-6}) \times 10^{-2}$	$(4.00013 \pm 5.69385 \times 10^{-6}) \times 10^{-2}$
.0004	$(2.00001 \pm 3.50259 \times 10^{-6}) \times 10^{-2}$	$(4.00001 \pm 4.02428 \times 10^{-6}) \times 10^{-2}$
.0008	$(2.00001 \pm 2.87772 \times 10^{-6}) \times 10^{-2}$	$(4.00001 \pm 3.32933 \times 10^{-6}) \times 10^{-2}$

Table 21: Confidence intervals for the OLS estimate of d when the data is generated with noise level $\nu_r = .01$.

δ	$d^* = .02$ ($N = 2048$)	$d^* = .04$ ($N = 4096$)
.0002	$(2.00004 \pm 2.41541 \times 10^{-5}) \times 10^{-2}$	$(4.00014 \pm 2.76640 \times 10^{-5}) \times 10^{-2}$
.0004	$(2.00000 \pm 1.68896 \times 10^{-5}) \times 10^{-2}$	$(4.00001 \pm 1.90853 \times 10^{-5}) \times 10^{-2}$
.0008	$(2.00003 \pm 1.40398 \times 10^{-5}) \times 10^{-2}$	$(4.00000 \pm 1.60390 \times 10^{-5}) \times 10^{-2}$

Table 22: Confidence intervals for the OLS estimate of d when the data is generated with noise level $\nu_r = .05$.

δ	$d^* = .02$ ($N = 2048$)	$d^* = .04$ ($N = 4096$)
.0002	$(2.00000 \pm 4.72903 \times 10^{-5}) \times 10^{-2}$	$(4.00014 \pm 5.48283 \times 10^{-5}) \times 10^{-2}$
.0004	$(2.00003 \pm 3.39327 \times 10^{-5}) \times 10^{-2}$	$(4.00002 \pm 3.87474 \times 10^{-5}) \times 10^{-2}$
.0008	$(2.00003 \pm 2.79911 \times 10^{-5}) \times 10^{-2}$	$(4.00003 \pm 3.19526 \times 10^{-5}) \times 10^{-2}$

Table 23: Confidence intervals for the OLS estimate of d when the data is generated with noise level $\nu_r = .1$.

δ	$d^* = .02$ ($N = 2048$)	$d^* = .04$ ($N = 4096$)
.0002	$(1.99272 \pm 0.000182978) \times 10^{-4}$	$(1.98142 \pm 0.000317616) \times 10^{-4}$
.0004	$(4.00035 \pm 0.000201885) \times 10^{-4}$	$(4.00737 \pm 0.000369841) \times 10^{-4}$
.0008	$(7.99833 \pm 0.000136586) \times 10^{-4}$	$(8.00332 \pm 0.000251291) \times 10^{-4}$

Table 24: Confidence intervals for the OLS estimate of δ when the data is generated with no noise (i.e., $\nu_r = 0.0$).

δ	$d^* = .02$ ($N = 2048$)	$d^* = .04$ ($N = 4096$)
.0002	$(1.99410 \pm 0.000958274) \times 10^{-4}$	$(1.98029 \pm 0.00111475) \times 10^{-4}$
.0004	$(4.00170 \pm 0.00108740) \times 10^{-4}$	$(4.00667 \pm 0.0012499) \times 10^{-4}$
.0008	$(7.99882 \pm 0.000800042) \times 10^{-4}$	$(8.00486 \pm 0.000923838) \times 10^{-4}$

Table 25: Confidence intervals for the OLS estimate of δ when the data is generated with noise level $\nu_r = .01$.

δ	$d^* = .02$ ($N = 2048$)	$d^* = .04$ ($N = 4096$)
.0002	$(1.99606 \pm 0.00475672) \times 10^{-4}$	$(1.98106 \pm 0.00541764) \times 10^{-4}$
.0004	$(4.00190 \pm 0.00524360) \times 10^{-4}$	$(4.01214 \pm 0.00593246) \times 10^{-4}$
.0008	$(7.99045 \pm 0.00391181) \times 10^{-4}$	$(8.00947 \pm 0.00444525) \times 10^{-4}$

Table 26: Confidence intervals for the OLS estimate of δ when the data is generated with noise level $\nu_r = .05$.

δ	$d^* = .02$ ($N = 2048$)	$d^* = .04$ ($N = 4096$)
.0002	$(2.00017 \pm 0.00932701) \times 10^{-4}$	$(1.97674 \pm 0.0107203) \times 10^{-4}$
.0004	$(4.00070 \pm 0.0105331) \times 10^{-4}$	$(4.01229 \pm 0.0120445) \times 10^{-4}$
.0008	$(7.99698 \pm 0.00778563) \times 10^{-4}$	$(8.00361 \pm 0.00886925) \times 10^{-4}$

Table 27: Confidence intervals for the OLS estimate of δ when the data is generated with noise level $\nu_r = .1$.

6 Conclusion

In this presentation, we have explored a “proof of concept” formulation of an inverse problem to detect and characterize voids or gaps inside of, or behind, a Dielectric medium. We have simplified the problem to one dimension and used the Maxwell equations to model a pulsed THz normally incident electromagnetic interrogating signal. We use Finite Element discretization in space, and Finite Differences in time, to simulate the electric field in the time domain. This is coupled with a Levenberg-Marquardt scheme in an optimization step with an innovative cost functional appropriate for reflected waves where phase differences can produce ill-posedness in the inverse problem when one uses the usual ordinary least squares criterion. We have successfully shown that it is possible to resolve gap widths on the order of $.2mm$ between a dielectric slab of $20cm$ and a metal (perfectly conducting) surface.

Future work on this problem will likely involve more efficient computational methods since currently the inverse problem involving a $20cm$ slab takes 10 hours. Further, more sophisticated models for describing the polarization mechanisms in non-homogenous materials must be developed. Finally, in order to take scattering and non-normally incident electromagnetic signals into account, multi-dimensional models will be necessary.

7 Acknowledgments

This research was supported (NLG) by the NASA Langley Graduate Student Researcher’s Program under grant NGT-1-02004 and in part (HTB) by the Air Force Office of Scientific Research under grant AFOSR F49620-01-1-0026.

References

- [1] B. M. Adams, H. T. Banks, J. E. Banks and J. D. Stark *Population Dynamics Models in Plant-Insect Herbivore-Pesticide Interactions*, CRSC Technical Report CRSC-TR03-12, North Carolina State University, August, 2003.
- [2] H. T. Banks, M. W. Buksas, and T. Lin *Electromagnetic Material Interrogation Using Conductive Interfaces and Acoustic Wavefronts*, SIAM: Frontiers in Applied Mathematics, Philadelphia, 2000.
- [3] R. L. Burden and J. D. Faires *Numerical Analysis*, PWS, Boston, 1993.
- [4] M. Davidian and D. M. Giltinan *Nonlinear Models for Repeated Measurement Data*, Chapman and Hall: Monographs on Statistics and Applied Probability 62, New York, 1995.
- [5] R. S. Elliott *Electromagnetics: History, Theory, and Applications*, IEEE Press, New York, 1993.
- [6] C. Johnson *Numerical Solution of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, New York, 1995.
- [7] C. T. Kelley *Iterative Methods for Optimization*, SIAM: Frontiers in Applied Mathematics, Philadelphia, 1999.